



Sistema para la supervisión remota de actividad física de pacientes mediante pulseras de sensores

Christian González García-Muñoz
Alejandro Huertas Herrero

Directores:
Javier Arroyo y Marlon Cárdenas

Curso 2017-2018

Trabajo de fin de grado del Grado en Ingeniería Informática

Facultad de Informática
Universidad Complutense de Madrid

Resumen

Este proyecto tiene como finalidad el desarrollo de un sistema para la supervisión remota de actividad física de pacientes de Parkinson. El sistema ha sido diseñado teniendo como objetivo ser lo menos intrusivo posible para el paciente para favorecer la comodidad de uso. El sistema está formado por cuatro elementos:

- Pulsera con sensores.
- Aplicación Android.
- Aplicación web de monitorización remota.
- Servidor con servicios API REST y HTTP.

Para llevar a cabo la supervisión del paciente nuestro sistema recoge datos sobre cómo éste realiza ciertas actividades, ya sean actividades cotidianas o alguna rutina de actividades marcada por un médico. Nuestro sistema soporta dos tipos de sensores ambos con conectividad BLE (Bluetooth de bajo consumo). De todos los servicios que ofrecen estos sensores nos hemos centrado en el acelerómetro y en el giroscopio ya que son los dos servicios que más información aportan sobre cómo realiza los movimientos el paciente y sobre cómo le afecta la enfermedad del Parkinson.

Para gestionar las pulseras y los datos generados por éstas hemos desarrollado una Aplicación Android. La aplicación almacena de forma local todos los datos generados por las pulseras mientras que se está grabando la realización de alguna actividad. Una vez los datos han sido grabados, la aplicación permite enviarlos a un servidor remoto haciendo uso de llamadas a un API REST. Además, la aplicación permite añadir recordatorios sobre tomas de medicamentos, que pueden resultar de gran ayuda al paciente, y permite grabar los minutos posteriores a la toma de la medicación para ver cómo afecta ésta a los pacientes.

Una vez los datos han sido enviados al servidor, la aplicación web de monitorización remota puede acceder a éstos haciendo uso del API REST mencionado antes. Esta aplicación web permite al médico supervisar de forma remota los avances del paciente mostrando los datos de una forma gráfica y amigable.

Todos los servicios incluidos en el servidor, el API REST, la base de datos MongoDB y el servidor HTTP son desplegados en forma de contenedores usando la herramienta open-source Docker.

Palabras clave: aplicación Android, sistema supervisión, Parkinson, bluetooth, Internet de las cosas, sensores, movimiento, actividades, medicamentos y visualización.

Abstract

The aim of this project is to develop a system for remote supervision about physical activities of Parkinson's patients. The system has been developed trying to be not too much intrusive for the patient, so its use is more comfortable. The system has four principal components:

- Sensors bracelet.
- Android application.
- Remote monitoring application web.
- Server with REST API and HTTP.

In order to supervise the patient, our system records data about how the patient performs certain activities. They can be daily activities or some routine table advised by a doctor. The system supports two sensors types, both are BLE (Bluetooth Low Energy) We have focused in accelerometer and gyroscope, which are the ones that give us more information about how the patient performs each movement.

To manage the bracelets and the data generated by them, we have developed an Android application. The application stores the data locally while some activity is being recording, when the data have been recorded, the data can be sent to a remote server using a REST API. The application also allows the user to add reminders for each medicine, which can be really useful for the patient, Moreover minutes after taking the medicine can be recorded with the sensors, so the doctor can see how medicine affects to the patient.

The web application can access to stored data using the REST API mentioned before. This application allows doctor to supervise the patient remotely. It shows the data in simple and friendly graphics.

All the infrastructure including the server, the REST API, the MongoDB and the HTTP server are deployed in containers using the open source project Docker.

Keywords: Android app, supervised system, Parkinson, bluetooth, Internet of Things, sensors, movement, activities, medicines y plotting.

Índice

| | |
|--|-----------|
| 1. Introducción | 7 |
| 1.1. Antecedentes | 7 |
| 1.2. Objetivos | 8 |
| 1.3. Plan de trabajo | 8 |
| 1.3.1. Fase 1: Documentación y estado del arte | 8 |
| 1.3.2. Fase 2: Desarrollo Aplicación Android | 8 |
| 1.3.3. Fase 3: Implementación de los servicios del servidor | 9 |
| 1.3.4. Fase 4: Visualización de datos | 9 |
| 2. Introduction | 10 |
| 2.1. Previous work | 10 |
| 2.2. Objectives | 10 |
| 2.3. Workplan | 11 |
| 2.3.1. Phase 1: Documentation and state of art | 11 |
| 2.3.2. Phase 2: Develop an Android App | 11 |
| 2.3.3. Phase 3: Implementation of the server services | 11 |
| 2.3.4. Phase 4: Data visualization | 11 |
| 3. Estado del arte | 12 |
| 3.1. Tipos de sensores usados | 12 |
| 3.2. Síntomas del Parkinson estudiados | 13 |
| 3.3. Libertad vs Actividades | 13 |
| 3.4. Innovación | 14 |
| 4. Descripción general del sistema | 15 |
| 5. Caso de uso | 17 |
| 6. Sensores | 18 |
| 6.1. Texas Instrument: CC2650 - Ultra-low power wireless MCU | 18 |
| 6.2. Hexiwear | 20 |
| 7. Aplicación Android | 23 |
| 7.1. Descripción de la funcionalidad | 23 |
| 7.2. Diagramas Casos de uso | 28 |
| 7.3. Diagramas de actividad | 29 |
| 7.4. Aspectos importantes de la implementación | 30 |
| 8. Aplicación web | 33 |
| 8.1. Descripción de la funcionalidad | 33 |
| 8.2. Diagramas de casos de uso | 36 |
| 8.3. Diagramas de actividad | 37 |
| 8.4. Aspectos importantes de la implementación | 37 |
| 9. Servicios del servidor | 38 |
| 9.1. API REST | 38 |
| 9.1.1. Detalles técnicos | 38 |
| 9.1.2. Esquema de datos | 38 |
| 9.2. Almacenamiento de datos | 38 |
| 9.3. Servidor HTTP | 39 |

| | |
|--|-----------|
| 9.3.1. Detalles técnicos | 39 |
| 9.4. Despliegue de la infraestructura | 39 |
| 9.4.1. Infraestructura sobre Docker | 39 |
| 9.5. Procesamiento de datos en el servidor | 40 |
| 10Conclusiones | 41 |
| 11Conclusions | 43 |
| 12Trabajo futuro | 44 |
| 12.1Aplicación Android | 44 |
| 12.2Aplicación Web | 44 |
| 12.3Estudio de los datos | 45 |
| 13Aportaciones | 46 |
| 13.1Christian González | 46 |
| 13.1.1Estado del arte | 46 |
| 13.1.2Aplicación Android | 46 |
| 13.1.3Aplicación web | 46 |
| 13.1.4API REST | 46 |
| 13.1.5Despliegue de la infraestructura | 47 |
| 13.1.6Procesamiento de datos | 47 |
| 13.1.7Otras aportaciones | 47 |
| 13.2Alejandro Huertas Herrero | 47 |
| 13.2.1Lectura de artículos | 47 |
| 13.2.2Aplicación Android | 47 |
| 13.2.3Aplicación Web | 48 |
| 13.2.4API Rest | 48 |
| 13.2.5Memoria | 48 |
| 13.2.6Organización | 48 |
| 14Apéndices | 49 |
| 14.1Esquema de datos API REST | 49 |
| 14.2Repositorios GitHub | 53 |

Índice de figuras

| | |
|---|----|
| 1. Sistema completo. | 15 |
| 2. Diagrama de interacción de los componentes del sistema. | 16 |
| 3. Sensor Hexiwear puesto. | 18 |
| 4. Sensor CC2650. | 18 |
| 5. Aspecto del sensor con la funda. | 19 |
| 6. Arquitectura interna sensor Texas Instrument. | 19 |
| 7. Sensor Hexiwar con correa. | 20 |
| 8. Diagrama del Hardware (Hexiwear). | 21 |
| 9. Pantalla principal de la aplicación. | 23 |
| 10. Actividades | 24 |
| 11. Añadir actividad | 24 |
| 12. Menú actividad | 24 |
| 13. Medicamentos | 25 |
| 14. Añadir medicamento | 25 |
| 15. Menú medicamento | 25 |
| 16. Enviar datos al servidor. | 27 |
| 17. Ayuda menú inicio. | 27 |
| 18. Aplicación | 28 |
| 19. Actividades | 28 |
| 20. Medicamentos | 28 |
| 21. Sensores | 28 |
| 22. Diagrama de activad grabar actividad. | 29 |
| 23. Arquitectura aplicación Android. | 30 |
| 24. Diagrama entidad-relación de la base de datos local. | 31 |
| 25. Pantalla principal de la aplicación web. | 33 |
| 26. Pantalla de actividades de un paciente de la aplicación web. | 34 |
| 27. Pantalla de detalle dato sensor de una actividad. | 34 |
| 28. Pantalla de medicación de un paciente de la Aplicación web. | 35 |
| 29. Caso de uso - Pantalla principal de la Aplicación web.. . . . | 36 |
| 30. Caso de uso - Pantalla de actividades de un paciente de la Aplicación web. 36 | |
| 31. Caso de uso - Pantalla de medicación de un paciente de la Aplicación web. 36 | |
| 32. Diagrama actividad - Ver datos de la actividad de un paciente. | 37 |
| 33. Diagrama de interacción de los contenedores de Docker. | 39 |

1. Introducción

La enfermedad de Parkinson es una enfermedad neurodegenerativa crónica que afecta al sistema nervioso central, y que progresa lentamente. Los síntomas del Parkinson afectan mayoritariamente a las habilidades motrices, y son bastante variados. Los principales síntomas son temblores, rigidez muscular, inestabilidad postural que hace que las personas afectadas pierdan el equilibrio fácilmente y bradicinesia, que afecta a la velocidad de los movimientos y que provoca que los movimientos simples del día a día se conviertan en algo difícil de realizar.

Como se puede observar los principales síntomas del Parkinson se pueden medir de alguna forma fijándonos en cómo se realizan algunos movimientos o en el caso del temblor fijándonos en la presencia o no de este movimiento. Esto hace que la enfermedad de Parkinson sea una enfermedad que se pueda ver ayudada por el gran avance de la tecnología IoT (Internet of Things). Esta tecnología está haciendo que cada vez sea más sencillo contar con sensores con suficiente calidad como para caracterizar los movimientos de estas personas con precisión y permitir su estudio.

El estudio de estos síntomas puede tener grandes ventajas en el diagnóstico previo y efectivo de la enfermedad, ya que en estados primarios los síntomas son puntuales y es difícil que se manifiesten justo cuando el paciente está pasando consulta en el médico. Además, en caso de ya estar diagnosticado, el estudio del movimiento permitiría a un médico contar con datos sobre cómo avanza la enfermedad, lo que ayudaría a la hora de seleccionar qué y cuánta medicación debe tomar el paciente.

Por todo lo anterior creemos que el desarrollo de un sistema como el nuestro cubriría con gran eficacia las necesidades mencionadas anteriormente. Además el hecho de usar sensores BLE permite abaratar el coste del sistema ya que tienen un precio bastante asequible y tienen un consumo de energía muy bajo y una alta fiabilidad. Además, dado que son sensores inalámbricos, son fáciles de llevar y poco intrusivos para el paciente.

1.1. Antecedentes

Este Trabajo de fin de grado está precedido por otro realizado en el curso anterior [16] en el cual nuestros compañeros implementaron un sistema capaz de reconocer la ejecución de actividades cotidianas usando una aplicación Android y sensores BLE similares a los usados en este proyecto, pero más difíciles de llevar, ya que en nuestro proyecto hemos priorizado la selección de sensores que sean cómodos de llevar para evitar que el paciente que los porte se sienta invadido por estos.

Nuestro proyecto trata de extender esta idea y aplicarla al problema específico del Parkinson. En lugar de reconocer tan solo actividades cotidianas, nuestro objetivo es ver como evoluciona la ejecución de estas actividades por pacientes diagnosticados de Parkinson y ayudar así a su supervisión. Este otro Trabajo de fin de grado estaba dirigido por los mismos profesores que han dirigido éste.

Además del trabajo de fin de grado citado en el párrafo anterior este trabajo viene precedido por numerosos artículos científicos en los que se intenta identificar los

síntomas del Parkinson mediante la monitorización de pacientes a través de sensores que éste lleva sujetos a alguna parte de su cuerpo. Se comentará en profundidad el alcance de estos artículos en la sección 3 de este documento.

1.2. Objetivos

El objetivo de este trabajo de fin de grado es la implementación de un sistema que permita supervisar la evolución de pacientes de Parkinson de la forma menos intrusiva posible. Para ello, grabaremos, a través sensores inalámbricos BLE, como aquéllos realizan actividades, ya sean actividades cotidianas o actividades pertenecientes a alguna rutina de ejercicios propuesta por un médico, para facilitar la identificación de los síntomas de dicha enfermedad. Después, proporcionaremos acceso a estos datos al propio médico para que pueda analizarlos e interpretarlos de forma totalmente accesible y sencilla.

Para ello se desarrollará una aplicación Android que permita gestionar la conexión bluetooth con los sensores BLE. Se intentará que la aplicación sea compatible con al menos dos tipos distintos de sensores, y almacenar toda la información captada por los sensores mientras que el paciente realiza las actividades, así como toda la información proporcionada por el paciente o la persona a su cargo que nos permita contextualizar cómo se realizaron estas actividades o si ocurrió algún problema durante la ejecución de la actividad. La aplicación permitirá también enviar los datos a un sistema de almacenamiento remoto haciendo uso de un API REST.

Una vez los datos recolectados se encuentran en el servidor éstos estarán accesibles a través de un aplicación web, también desarrollada por nosotros, que hará uso del mismo API REST que la aplicación Android para recuperar los datos recolectados, y que permitirá al medico supervisar la evolución del paciente de forma continuada y remota sin necesidad de establecer una cita para que el paciente o la persona a su cargo le proporcione los datos recolectados.

1.3. Plan de trabajo

El plan de trabajo del proyecto cuenta con cuatro fases explicadas a continuación:

1.3.1. Fase 1: Documentación y estado del arte

Durante esta primera fase, el objetivo es documentarnos lo máximo posible y ponernos en contexto sobre el estado del arte actual en el problema de la monitorización mediante sensores de enfermos de Parkinson. Para ello los directores nos proporcionarán toda la información necesaria sobre el TFG previo comentado anteriormente y artículos con relación a esta temática que leeremos con la intención de ver qué soluciones podrían ser compatibles con la nuestra y encontrar posibles mejoras.

1.3.2. Fase 2: Desarrollo Aplicación Android

En esta segunda fase se realizará el desarrollo de la aplicación Android que nos permitirá recoger la información de los sensores así como del contexto del paciente (medicación, actividades,...) para su posterior uso.

1.3.3. Fase 3: Implementación de los servicios del servidor

Una vez implementada la aplicación Android capaz de recopilar los datos y almacenarlos localmente en el teléfono, implementaremos los servicios necesarios para ser capaces de extraer esta información del teléfono y enviarla al servidor para que sea procesada.

1.3.4. Fase 4: Visualización de datos

Una vez implementada la aplicación que nos permite recopilar la información y la infraestructura necesaria para manipular esta información fuera del teléfono móvil del paciente implementaremos una aplicación web que servirá para visualizar los datos de una forma amigable para el médico que deba interpretar la información.

2. Introduction

The Parkinson disease, is a chronic neurodegenerative disease that affects the central nervous system and advances slowly. The symptoms affect mostly to the motor habilities. The main symptoms are tremors, muscular stiffness, postural instability which make the people who suffer it to overbalance easily and bradykinesia which affects speed of the movements and which turn the easy movements into something hard.

It is easy to see that the main Parkinson's symptoms can be measured looking at the way some movements are executed or in case of the tremor just examining its presence. This makes that the study of the Parkinson disease can be helped by the great advancements made at IoT (Internet of Things) technology which makes easier to deal with sensors with enough accuracy and quality to allow the study of these movements

For all the previous reasons, we believe that the development of a system like ours will cover the needs mentioned above. Besides the use of BLE sensors allows to decrease the costs of the systems and provide a high reliability and a very low consume. And because they are wireless they are so comfortable to wear for the patient.

2.1. Previous work

This project is preceded by another one made the previous year [16] in which our colleges develop a system capable of recognize common activities using an Android app and BLE sensors like ours but more difficult to wear because our project prioritize to select wearables sensors.

Our project apply the same idea to the specific problem of the Parkinson disease, instead of just recognize common activities our target is to observe how the execution of these activities in Parkinson patients evolves and help the supervision of these activities. This previous project was directed by the same professors than ours.

Additionally our project is preceded by many scientific articles which try to identify Parkinson symptoms using wearables sensors. These articles will be discussed deeply at the section 3 of this document

2.2. Objectives

Once all the collected data are at the remote server they will be available through a web app, also developed by us, which will use the same REST API of the Android app for get the collected data and which will allow the continuous supervise of the patient by the doctor without need any date for get the data.

2.3. Workplan

The project work plan has four phases each one explained below:

2.3.1. Phase 1: Documentation and state of art

During these first phase the objective is to documentate the current state of the art of the monitoring using wearables sensors in Parkinson patients. In order to achive it ours director will provide us with scientific articles related with this subject and all the information about the previos project mentioned above.

2.3.2. Phase 2: Develop an Android App

In this second phase we develop an Android app which allows us to collect the sensors information and the patient context for it's later use.

2.3.3. Phase 3: Implementation of the server services

Once the Android app is capable to collect the sensor data and store it locally we will implement the services in order to be able to extract these data from the smartphone and send it to the remote server.

2.3.4. Phase 4: Data visualization

Once the app is capable of stores the sensor data and we have al the services we develop a web app which allows to visualize the data in order the doctor can analyze it.

3. Estado del arte

“La enfermedad de Parkinson (EP), también denominada mal de Parkinson, parkinsonismo idiopático, parálisis agitante o simplemente párkinson, es una enfermedad neurodegenerativa crónica caracterizada por bradicinesia (movimiento lento), rigidez (aumento del tono muscular) y temblor.” [6]

Uno de los artículos que mejor puede servir como introducción para ver cuáles son las necesidades, acerca del diagnóstico y la supervisión de enfermos de Parkinson usando sensores “wearables”, y hasta qué punto han sido cubiertas o estudiadas es [2], en este artículo se habla sobre el creciente interés que surge alrededor del estudio de los distintos síntomas del Parkinson usando tecnología “wearable” y de cómo el uso de esta tecnología, permitiría recolectar datos de forma poco intrusiva, objetiva y ecológica para obtener diagnósticos más precisos y menos sujetos a la subjetividad del paciente o a cómo éste expresa los síntomas sufridos, ya que muchas veces durante las consultas médicas los síntomas no se muestran y el médico no puede evaluarlos por sí mismo.

“Specifically, quantitative assessments using wearable technology may allow for continuous, unobtrusive, objective, and ecologically valid data collection.”

Además, el artículo nos hace un recorrido sobre los distintos síntomas del Parkinson explicando cómo el uso de la tecnología wearable podría ayudar a trabajar sobre cada uno de ellos. El artículo también habla sobre la necesidad de identificar la existencia de sedentarismo y evitarla en pacientes de Parkinson para poder proporcionar un buen nivel de vida.

Nuestro sistema cubre gran parte las necesidades enumeradas en este artículo, ya que se puede adaptar al estudio de los distintos síntomas del Parkinson enumerados en el artículo dado que está diseñado para ser un sistema genérico e independiente de la posición de los sensores y del tipo de actividad que se realiza. Incluso cubre la necesidad de identificar el sedentarismo ya que se le puede solicitar al paciente que grave distintas actividades de su vida cotidiana para ver el nivel de actividad física que realiza en su día a día.

3.1. Tipos de sensores usados

A la hora de seleccionar el tipo de sensores a utilizar, todos los artículos que hemos leído usan, ya sea en solitario o acompañado por otros tipos de sensores, acelerómetros triaxiales. En [3] realizan una comparación entre los datos recogidos por un acelerómetro y un podómetro para estudiar la congelación de pisada o FOG por sus siglas en inglés, en este artículo hacen que tanto pacientes de parkinson como persona que no padecen esta enfermedad caminen durante longitudes variables y después entrenan algoritmos de aprendizaje automático con estos datos para intentar obtener el número de pasos que se han realizado. Una de las conclusiones del artículo es que los datos obtenidos por el acelerómetro ofrecen un mejor resultado que los obtenidos por el podómetro para longitudes superiores a 5 metros.

Otro artículo en el que se habla sobre el uso de acelerómetros para monitorizar a pacientes de Parkinson es [4], en este artículo se estudia la factibilidad de usar acelerómetros para identificar la gravedad de discapacidades motoras. En concreto temblores, bradicinesia, y discinesia, llegando a obtener unas tasas de error bastante bajas. Para ello, sitúan sensores en varias partes del cuerpo y recogen los datos. Estos datos se usan después para entrenar un algoritmo de aprendizaje automático (Support Vector Machine) cuyo resultado se compara con el diagnóstico realizado por un médico. Una de las limitaciones que vemos en este artículo, y que no tendremos en este proyecto, es el uso de sensores alámbricos que son mucho mas intrusivos para el paciente que los lleva.

Tras la lectura de los artículos [3] y [4] entre otros, podemos concluir que el uso de acelerómetros como sensores para realizar este proyecto es bastante acertado ya que en la mayoría de estudios obtienen resultados satisfactorios. Además, en nuestro proyecto, usaremos giroscopios para apoyar la tarea de los acelerómetros e intentar mejorar los resultados.

3.2. Síntomas del Parkinson estudiados

Como se cuenta en [2] el Parkinson tiene diferentes síntomas, algunos motores, la gran mayoría, y algunos no motores.

En este proyecto nos centraremos en los síntomas motores, la mayoría han sido estudiados ya de alguna forma, por ejemplo en [3] se centran en el estudio de la congelación de pisada [9], [5] se centra en el estudio de los temblores o [4] en el que se estudian varios sintomas, en este en concreto se estudian temblores, bradicinesia [7] y discinesia [8].

Nuestro sistema puede usarse para recopilar datos de todos los síntomas motores que tiene el Parkinson ya que las diferencias entre éstos suele ser en las posiciones seleccionadas para poner los sensores o en las actividades que realiza el paciente mientras que se le está monitorizando y nuestro sistema ofrece una libertad total respecto a estas opciones.

3.3. Libertad vs Actividades

En [2] se habla sobre la monitorización continua y no intrusiva de pacientes Parkinson, aun así, en la gran mayoría de los artículos que hemos leído la monitorización de enfermos de Parkinson, ya sea en un entorno controlado o no, suele ir ligada a la realización repetitiva de distintas actividades que ayudan a caracterizar los síntomas que se estén estudiando en cada caso. Nosotros pensamos que la realización de estas tareas en sí, pese a que aportan información muy valiosa sobre cómo evoluciona el paciente, son intrusivas, ya que obligas al paciente a realizarlas de forma periódica y es necesario contar con la colaboración de éste. Por este motivo, uno de los objetivos a largo plazo es que se pueda monitorizar al paciente realizando tareas del día a día con total libertad.

Nuestro sistema tiene en mente estas necesidades y está diseñado para ser suficientemente genérico y permitir, tanto la grabación del paciente realizando tareas establecidas por un médico o el encargado del estudio de éste, como tareas de su día a día.

3.4. Innovación

Una de las características que tiene nuestro sistema y la cual no hemos visto en ninguno de los artículos leídos, es que nuestro sistema no solo se encarga de monitorizar y almacenar los datos, sino que también es capaz de enviarlos a un servidor remoto y ofrece una interfaz, en forma de aplicación web, para que el encargado de analizar los datos, normalmente el médico pueda analizarlos y trabajar sobre ellos de forma remota sin tener que estar con el paciente mientras éste realiza los datos o tener que concertar citas para extraer los datos del sistemas pertinente.

Creemos que esta funcionalidad cierra el ciclo del sistema y lo hace totalmente independiente e idóneo para la tarea para la que está diseñado además, nuestro sistema ofrece mucha facilidad en caso de querer añadir funcionalidades, como por ejemplo aplicar algoritmos de aprendizaje automático sobre los datos ya que siempre se puede acceder a los datos haciendo uso del API REST y a partir de ahí implementar la nueva funcionalidad.

4. Descripción general del sistema

Nuestro sistema, consta de 4 partes principales, **sensores**, **aplicación Android**, **aplicación web** y **servidor**, las cuales se encargan de recolectar, almacenar, gestionar y representar los datos sobre las actividades de los pacientes de Parkinson respectivamente.

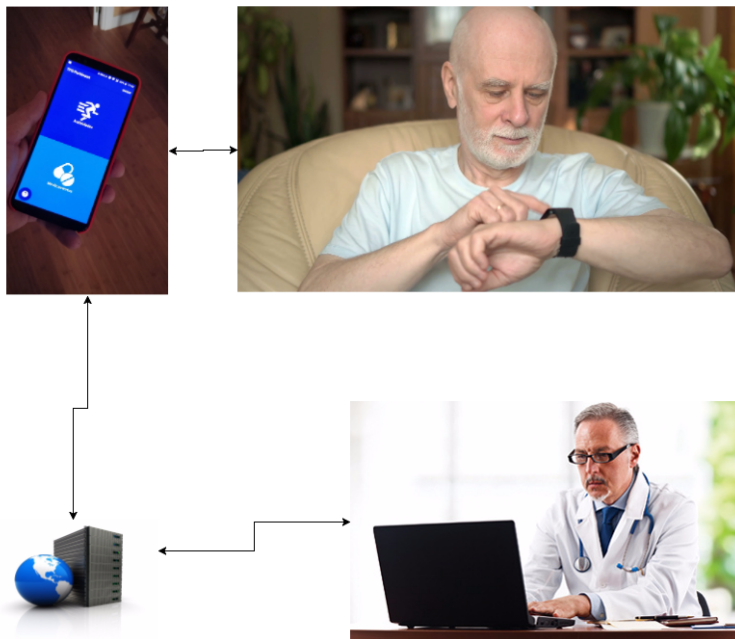


Figura 1: Sistema completo.

Para la recolección de los datos se hace uso de **sensores BLE**, en concreto de sus servicios de acelerómetro y giroscopio. A la hora de seleccionar los sensores usados en este proyecto se ha tenido en cuenta que tanto su tamaño como su peso debe ser lo suficientemente reducido para que sean fáciles de portar mientras se realizan las actividades que se desean grabar.

La **aplicación Android** se encarga de gestionar la conexión con los sensores BLE y de almacenar de forma local todos los datos que éstos van enviando así como toda la información extra proporcionada por el paciente o la persona a su cargo. Esta información facilita el posterior análisis de los datos por el propio médico. Además la aplicación Android se encarga de enviar los datos recolectados, correctamente etiquetados para identificar al paciente y al servidor remoto. La aplicación también permite establecer recordatorios sobre la toma de la medicación para el paciente y grabar los minutos después a la toma de esta medicación ya que suelen ser minutos bastante interesantes para evaluar cómo afecta la medicación al paciente.

Para realizar la **comunicación** entre la **aplicación Android** y el **servidor** se hace uso de un API REST desarrollada por nosotros utilizando el framework de Python Eve. Este API nos permite abstraernos del sistema de almacenamiento que hay detrás y proporciona una interfaz estándar y sencilla.

En el **servidor** los datos recogidos a través del API REST se almacenan en una

base de datos MongoDB.

Una vez los datos recolectados están en la base de datos del servidor, estarán accesibles para la aplicación web encargada de mostrar los datos de una forma visual al médico para que éste pueda supervisar la evolución del paciente. Esta aplicación web hace uso del API REST mencionado anteriormente para recuperar los datos del servidor.

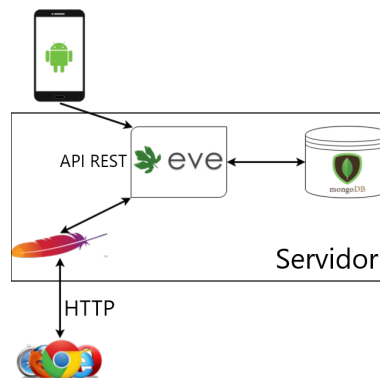


Figura 2: Diagrama de interacción de los componentes del sistema.

5. Caso de uso

En el siguiente texto se cuenta, a modo de historia, cómo sería un posible caso de uso de nuestro sistema, tanto por parte del paciente como del médico.

Pepe, como cada mes, acude a visitar a su médico para ver cómo está evolucionando el Parkinson. Lleva sufriendo la enfermedad unos tres años y hasta la fecha el único control que tenía era la visita a su médico. Ahora la enfermedad comienza a avanzar más rápido y su médico necesita tener un control más continuo y objetivo de su evolución. Para ello, deciden probar una nueva aplicación que le habían presentado dos alumnos de la facultad de informática de la universidad Complutense de Madrid.

En la visita, el médico le explicó a Pepe cómo iban a funcionar a partir de ahora. Le enseñó la aplicación un poco por encima, para que se fuera familiarizando con ella. Le dijo que a partir de ahora debería hacer dos cosas al día. En primer lugar, le puso una tabla de ejercicios, esta tabla debería seguirla a diario y cada vez que fuera a hacerla debería apuntar en la aplicación la actividad y grabarla con los sensores. Por otro lado, debería apuntar en la aplicación toda la medicación que se estaba tomando y de esos medicamentos, dos, debería grabar la actividad una vez se hubiera tomado el medicamento. A Pepe le quedó todo bastante claro y estaba bastante emocionado con utilizar la aplicación.

Cuando Pepe llegó a su casa, apuntó en la aplicación todos los medicamentos, era bastante sencillo, solo tenía que anotar el nombre, los días y la hora, la aplicación se encargaba de recordarle cada día la toma de cada pastilla. Al día siguiente, comenzaría con la tabla de ejercicios.

A la mañana siguiente, Pepe comenzó con la tabla de ejercicios. El primero de ellos consistía en andar durante 10 minutos. Registró la actividad en la aplicación y a continuación, conectó los sensores. Una vez terminó con esa actividad, hizo las demás de la tabla, siguiendo el mismo procedimiento. Cuando terminó, le dio al botón de enviar, de tal forma que el médico pudiera consultar todos los datos. Después de comer, se tenía que tomar varias pastillas, la aplicación se lo recordó y Pepe se tomó todos los medicamentos y grabó la actividad después de dos de ellos, como le había dicho el médico.

Pepe repitió esta operación durante todos los días. Mientras tanto, su médico iba revisando los datos periódicamente en su ordenador. Dos veces durante la semana se metía en la aplicación web para consultar los datos de Pepe. En ella podía ver gráficas con los datos de los sensores. Viendo las gráficas pudo darse cuenta de que uno de los medicamentos no estaba funcionando bien, pues las gráficas mostraban datos extraños justo después de la toma de ese medicamento. También, pudo darse cuenta de que Pepe no se tomaba uno de los medicamentos que se tenía que tomar por la noche, se le solía olvidar con mucha frecuencia.

En la consulta siguiente, el médico le comentó todo lo que había ido viendo en la aplicación. Decidió cambiarle la medicación que parecía no estar haciendo efecto y le cambió la hora de la toma de la medicación que se le solía olvidar con frecuencia.

6. Sensores

Durante el trabajo hemos utilizado dos tipos de sensores de las marcas Texas Instrument y Hexiwear, que se presentan a continuación. Ambos sensores están pensados para llevar como pulsera, aunque se podrían poner en otra parte del cuerpo si fuera necesario. Aquí puede verse un ejemplo de cómo quedaría uno de ellos en la muñeca.



Figura 3: Sensor Hexiwear puesto.

El teléfono no es necesario que esté siempre con el paciente. Para recoger datos basta con conectar los sensores al teléfono, toda la conexión es totalmente inalámbrica, a diferencia de la mayoría de artículos que utilizaban sensores conectados con cables. La aplicación desarrollada en Android, puede trabajar con ambos sensores de la misma forma, lo único que se necesita es buscar el dispositivo en la lista que aparece, seleccionarlo y a continuación se realizará la conexión. Todo el proceso es transparente al usuario, de tal forma que simplifica el uso de los sensores para el usuario. También es importante mencionar que la aplicación es capaz de trabajar con varios sensores a la vez, de forma que recoge datos de todos ellos. Aunque esta funcionalidad sólo está pensada para los sensores de Texas Instruments, ya que los de Hexiwear al tener pantalla incorporada, tiene más sentido que sólo se use uno cada vez.

6.1. Texas Instrument: CC2650 - Ultra-low power wireless MCU

Descripción

El dispositivo es un sensor BLE (Bluetooth Low Energy) es decir, es bluetooth de bajo consumo. Cuenta con varios sensores dentro del propio dispositivo. Se puede conectar con cualquier dispositivo que disponga de bluetooth. En nuestro caso se ha utilizado conectándolo con un teléfono móvil.



Figura 4: Sensor CC2650.



Figura 5: Aspecto del sensor con la funda.

Hardware

El dispositivo contiene un procesador ARM de 32 bits que funciona a 48 MHz, el cual se encarga de controlar el funcionamiento de todos los sensores que lo forman. Los servicios disponibles son los siguientes:

- Sensor de temperatura.
- Sensor de movimiento (acelerómetro y giroscopio)
- Sensor de humedad.
- Sensor de presión.
- Sensor óptico.

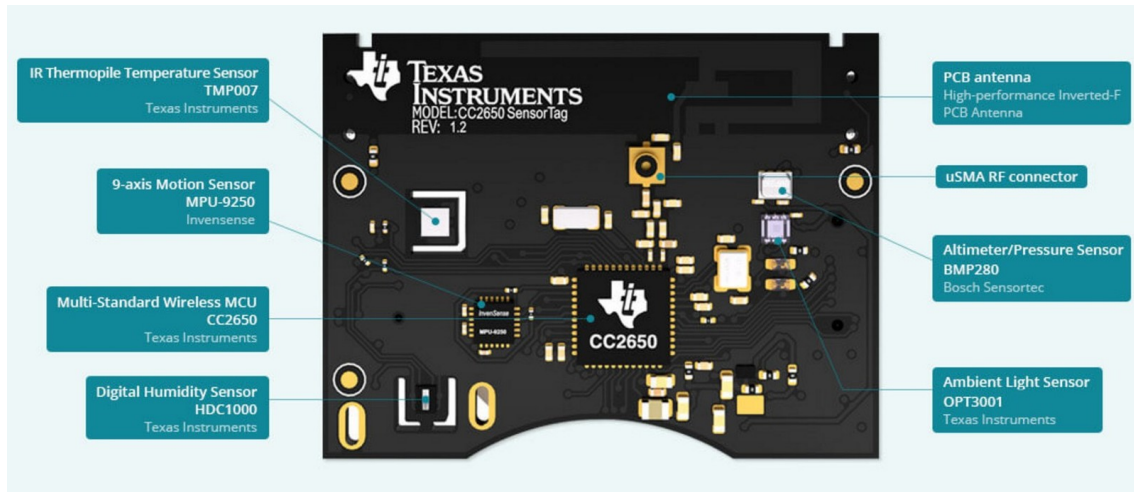


Figura 6: Arquitectura interna sensor Texas Instrument.

Software

Para interactuar con el sensor se puede utilizar cualquier dispositivo que tenga bluetooth, desde un dispositivo móvil hasta un ordenador. Existen librerías en Python que facilitan la interacción con el mismo. Debido a que hemos conectado los sensores con un teléfono móvil con Android, estas librerías no han sido usadas, pues usar Python en el terminal habría sido demasiado costoso y poco eficiente. Para acceder a los sensores se utilizan direcciones físicas, conocidas como UUID (Universally Unique Identifier) También hay UUID que permiten configurar el sensor para cambiar parámetros como la frecuencia de funcionamiento, qué sensores deben activarse, etc...

Integración con nuestra aplicación Android

La integración con nuestra aplicación se realiza a través de la propia aplicación, sin el uso de aplicaciones externas. Nuestra aplicación permite recoger datos de todos sus sensores (acelerómetro, giroscopio y magnetómetro) Una vez se conecta, se empiezan a recoger datos que son almacenados en la base de datos local. Todos esos datos pueden ser enviados posteriormente al servidor remoto, momento en el cual son borrados de la base de datos local.

6.2. Hexiwear

Descripción

Es un dispositivo pensado para IoT (Internet de las cosas) bastante pequeño y con un diseño amigable que permite usarlo como reloj inteligente. Hexiwear es un sensor bastante versátil completamente software libre.



Figura 7: Sensor Hexiwar con correa.

Harwdare

El dispositivo cuenta con un procesador MCU Kinetis K64x de NXP (ARM Cortex-M4, 120 MHz, 1M Flash, 256K SRAM), BLE NXP Kinetis KW4x (ARM® Cortex-M0+, Bluetooth Low Energy & 802.15.4 Wireless MCU), batería de litio, pantalla OLED de 1.1", interfaz táctil y un led RGB. Además cuenta con los siguientes sensores:

- Acelerómetro tridimensional.
- Magnetómetro tridimensional.
- Giroscopio triaxial.
- Sensor de presión.
- Sensor de pulsaciones.
- Sensor de humedad
- Sensor óptico

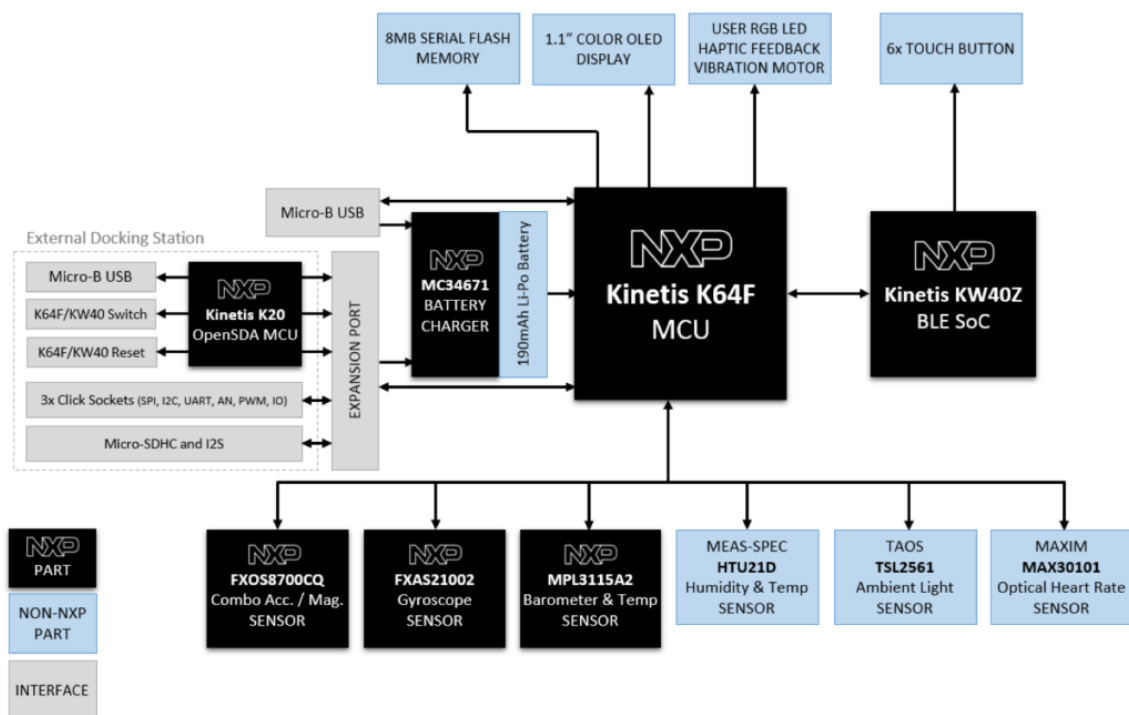


Figura 8: Diagrama del Hardware (Hexiwear).

Software

Hexiwear ofrece aplicaciones tanto en iOS como en Android para interactuar con sus sensores. Además ofrece una nube a través de la cual recuperar los datos del sensor haciendo uso de un API REST. Dado que nuestra aplicación está diseñada para ser capaz de monitorizar a un paciente en cualquier momento, no podíamos hacer uso de la nube ya que necesita conexión a Internet y tampoco de la aplicación que

proporcione Hexiwear. Por lo que tuvimos que implementar en nuestra aplicación la integración con el sensor haciendo uso de las librerías BLE proporcionadas por Android.

Integración con la aplicación Android

Como se menciona en el apartado anterior la interacción con el sensor se hace a través de nuestra propia aplicación Android. Mientras el dispositivo está conectado con la aplicación se recogen los datos de acelerómetro y giroscopio y se almacenan en la base de datos local de la aplicación para posteriormente enviarlos y procesarlos en el servidor remoto.

7. Aplicación Android

7.1. Descripción de la funcionalidad

La aplicación ha sido desarrollada para dispositivos Android, con una versión igual o superior a Android 4.4 (KitKat). Nuestra aplicación se caracteriza por: diseño intuitivo y facilidad de uso, ya que está pensada para personas que pueden tener temblores mientras la usen o pueden encontrarse en una avanzada edad. Toda la recogida de datos de los sensores está automatizada, el paciente sólo tiene que conectar la aplicación con cada sensor y puede olvidarse del teléfono en ese momento. La aplicación es lo menos invasiva posible, para ello tan sólo cuenta con notificaciones cuando se tiene que tomar las pastillas. Cada notificación indica la pastilla que debe tomarse y cuenta con dos opciones, de esta forma se puede llevar un control de las pastillas que se ha tomado. La aplicación permite la conexión asíncrona con el servidor para el almacenamiento permanente de los datos.

La aplicación ofrece las dos siguientes funciones principales:

- **Bitácora de actividades:** cuenta con una lista de todas las actividades que realiza el paciente. Cada vez que se añade una nueva se graba la actividad con los sensores. En todo el sistema en general, esta es la parte que permite obtener los datos del sensor.
- **Bitácora de medicamentos:** cuenta con una lista de toda la medicación que tiene que tomar el paciente, esta lista se puede modificar siempre que el paciente quiera. La aplicación le notifica al paciente cada vez que se tiene que tomar un medicamento y le da la posibilidad de grabar la actividad después de la toma. En todo el sistema en general, ésta es la parte que permite recoger datos de la toma del medicamento, así como del efecto que causa en el paciente.

A continuación, se muestran capturas de la aplicación y cómo funciona cada apartado de la misma.

Menú inicio

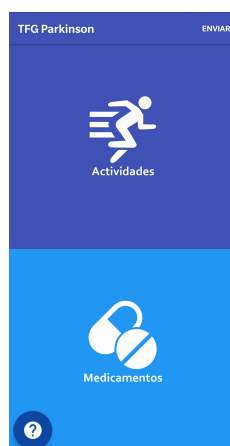


Figura 9: Pantalla principal de la aplicación.

Esta es la pantalla que aparece cuando se inicia la aplicación. Podemos ver dos grandes botones, el primero de ellos permite acceder a todo lo relacionado con las actividades y el segundo a todo lo relacionado con los medicamentos. Para iniciar la aplicación, no es necesario tener los sensores a mano ni conectados a la aplicación. De esta forma, la aplicación puede ser usada simplemente para consultar datos y no para recogerlos.

Puede verse como en el menú superior hay un botón. Este botón permite enviar los datos al servidor remoto. El botón sólo aparece cuando hay datos disponibles que enviar. Mientras no los haya, el botón no aparece. De esta forma el paciente sabe en todo momento si tiene algo que enviar.

Gestión de las actividades

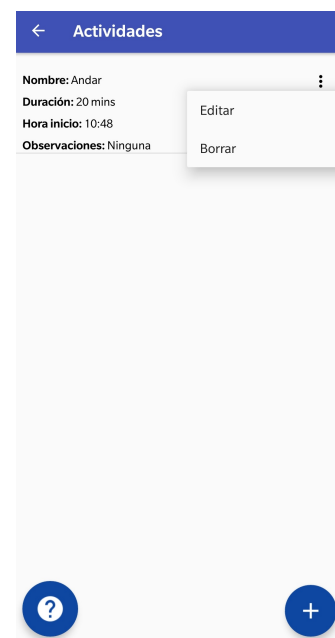
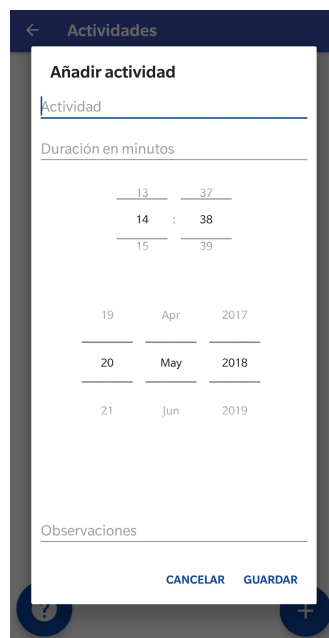


Figura 10: Actividades Figura 11: Añadir actividad Figura 12: Menú actividad

Las tres pantallas de arriba muestran cómo se trabaja con las actividades. La pantalla principal cuenta con una lista de todas las actividades que se realizan, a modo de histórico y con un botón que permite añadir una nueva actividad al sistema.

La información que se guarda de cada actividad es:

- **Nombre** de la actividad.
- **Duración** de la actividad.
- **Hora y fecha** en las que se realizó la actividad.
- **Observaciones** se trata de un texto libre en el que el paciente puede indicar cualquier información relevante ocurrida durante la realización de la actividad, o si ocurrió algo que impidió que la realizara con normalidad.

Una vez se añade la actividad, la aplicación pedirá al paciente que conecte los sensores para comenzar a grabar los datos de la actividad. Una vez estén conectados, aparecerá una pantalla que muestra los datos emitidos por los sensores en tiempo real, con un botón de parar, que detendrá la recogida de datos.

Al pulsar en el botón de añadir actividad (botón flotante de abajo a la derecha en la Figura 10) se nos abre un cuadro de diálogo en el que podemos rellenar cada uno de los campos. El campo de actividad se ha dejado libre, aunque lo más normal es que el paciente introduzca actividades cotidianas o de su rutina de ejercicios.

Por último, cómo se ve en la figura 12, para cada actividad se puede desplegar un menú contextual a través del icono de los "Tres puntos" en el que tenemos dos opciones disponibles: editar y borrar. Editar abrirá un cuadro de diálogo como el de añadir, pero con todos los campos rellenos con la información de dicha actividad (aparecerán bloqueados todos los campos rellenos menos las observaciones) Borrar abrirá un cuadro de diálogo preguntando si de verdad se desea borrar la actividad, en caso afirmativo la actividad será borrada de la lista que contiene todas las actividades.

Gestión de los medicamentos

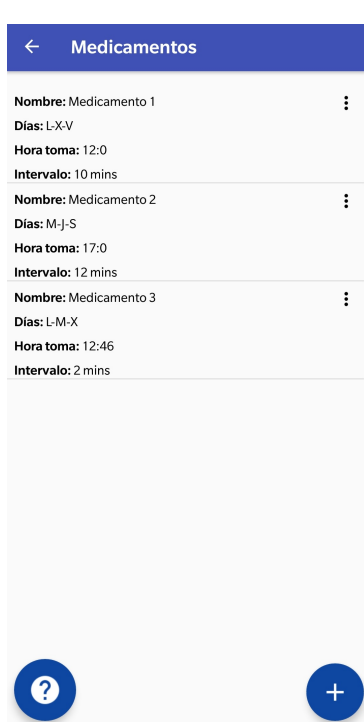


Figura 13: Medicamentos

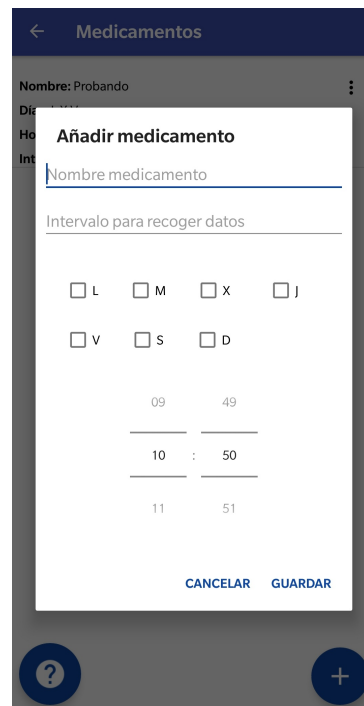


Figura 14: Añadir medicamento

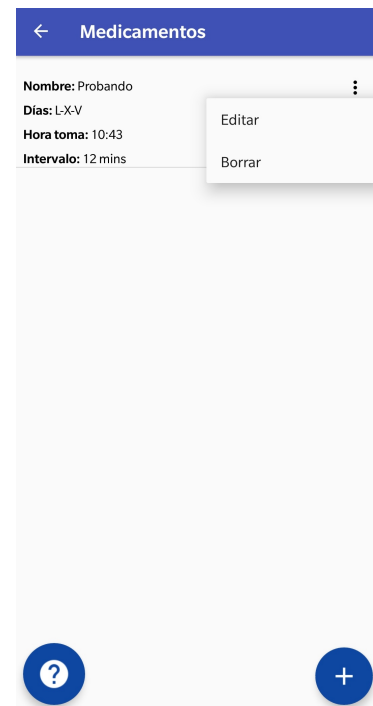


Figura 15: Menú medicamento

La gestión de los medicamentos está formada por dos funcionalidades. En primer lugar se permite añadir medicamentos y así poder tener una lista que se lea mejor con todos los medicamentos que se toma el paciente. Cada vez que se añade un medicamento, se programa una notificación, la cual saltará en el momento de la toma. En

ese momento el paciente puede indicar que se ha tomado la pastilla o que no. De esta forma se tiene un histórico de las tomas del paciente. En segundo lugar, si el paciente indica que se lo ha tomado, se le dará la posibilidad de grabar con los sensores una actividad, para de esta forma, recoger el efecto de la medicación.

Las tres pantallas de arriba muestran cómo se trabaja con los medicamentos. La pantalla principal, Figura 13, cuenta con una lista de todos los medicamentos que tiene que tomar el paciente y con un botón que permite añadir un nuevo medicamento al sistema.

La información que se guarda de cada medicamento es:

- **Nombre** del medicamento.
- **Intervalo** para la recogida de datos. Este parámetro indica los minutos después de la toma del medicamento, en los que los datos recogidos por los sensores podrían verse alterados debido al efecto de la medicación.
- **Días de la semana** en los que se tiene que tomar la medicación.
- **Hora** a la que se tiene que tomar la medicación.

Al pulsar en el botón de añadir medicamento (botón flotante de abajo a la derecha Figura 13) se nos abre un cuadro de diálogo en el que podemos rellenar cada uno de los campos.

Por último, como se ve en la Figura 15, para cada medicamento se puede desplegar un menú contextual a través del icono de los "Tres puntos" en el que tenemos dos opciones disponibles: editar y borrar. Editar abrirá un cuadro de diálogo como el de añadir, pero con todos los campos rellenos con la información de dicho medicamento (aparecerán bloqueados aquellos que no se puedan editar, es decir, el nombre) Borrar abrirá un cuadro de diálogo preguntando si de verdad se desea borrar el medicamento, en caso afirmativo el medicamento será borrado de la lista que contiene todos los medicamentos.

Envío de datos al servidor remoto

El botón de enviar que se encuentra situado arriba a la derecha en la pantalla principal (ver Figura 9), permite enviar todos los datos que aún no han sido enviados al servidor, de forma que no se tendrán datos repetidos en el servidor. Si algo falla durante el envío, los datos podrán ser enviados en otro momento. Se enviarán al servidor:

- Las actividades realizadas, incluyendo los datos recogidos por los sensores.
- Los medicamentos que se tiene que tomar el paciente.

Una vez pulsamos el botón, se abre el cuadro de progreso que puede verse en la imagen. Desaparece una vez todos los datos han sido enviados. Después se informa al usuario de si ha ido todo bien o ha fallado algo.

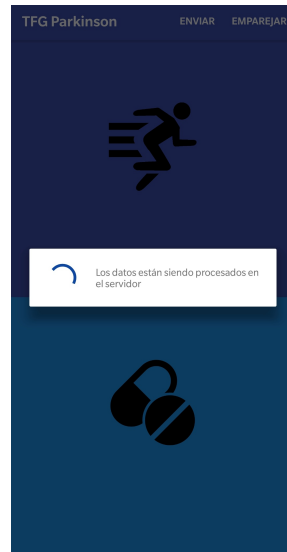


Figura 16: Enviar datos al servidor.

Ayuda

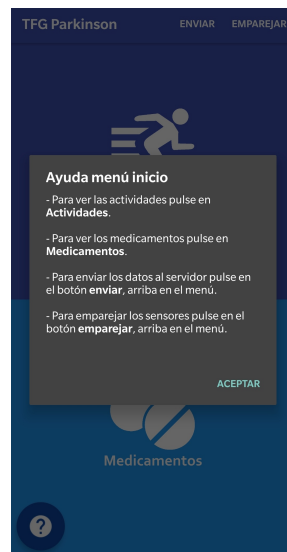


Figura 17: Ayuda menú inicio.

En todas las pantallas de la aplicación puede verse un botón flotante abajo a la izquierda, es el botón de ayuda. Puesto que la aplicación está dirigida a pacientes de Parkinson, en su gran mayoría tienen avanzada edad, consideramos que era buena idea incluir un botón que muestre información acerca de cómo se realiza cada acción.

El botón de ayuda, muestra en cada pantalla cómo realizar todas las acciones permitidas en dicha pantalla. Como podemos ver en este caso muestra cómo acceder a cada apartado, así como enviar datos y conectar los sensores. Resalta en negrita las partes más importantes, para que puedan ser vistas de forma sencilla y rápida.

7.2. Diagramas Casos de uso

A continuación se muestran los diagramas de casos de uso correspondientes a la aplicación web:

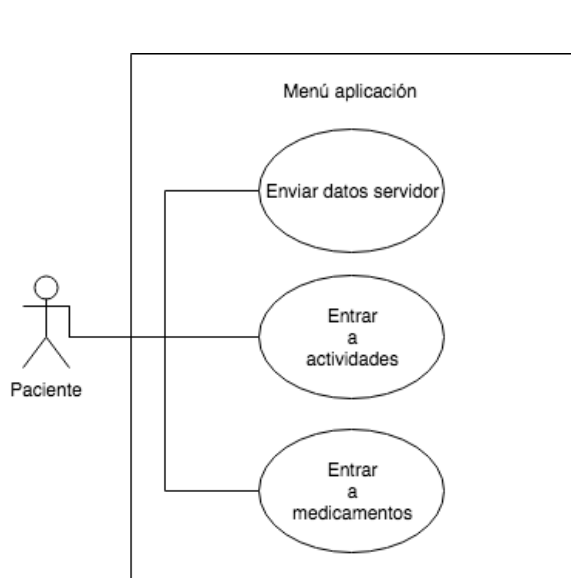


Figura 18: Aplicación

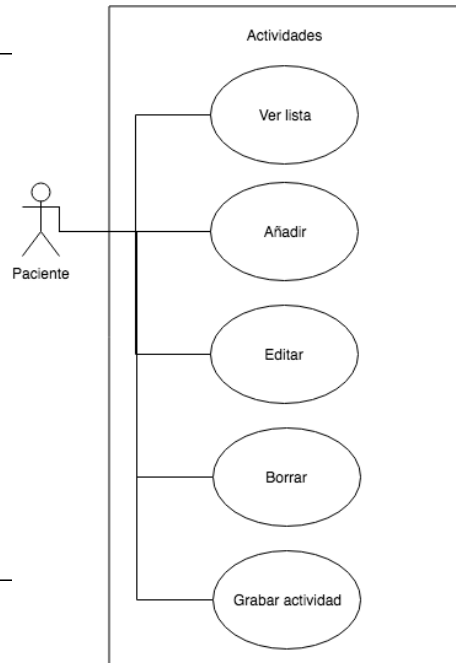


Figura 19: Actividades

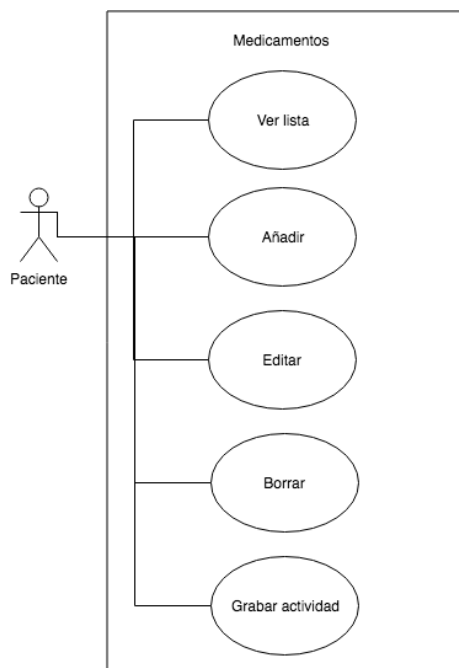


Figura 20: Medicamentos

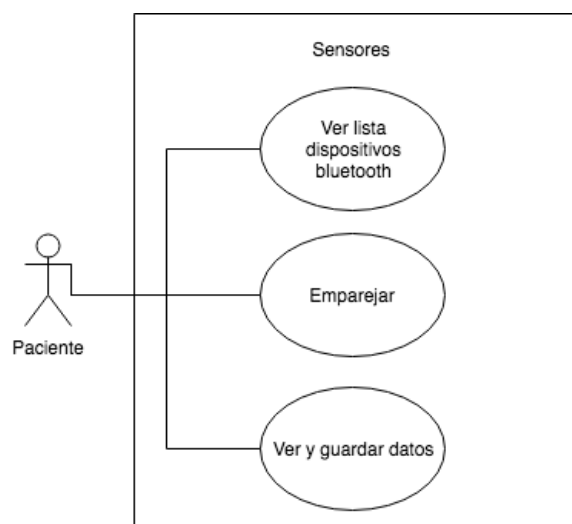


Figura 21: Sensores

7.3. Diagramas de actividad

A continuación, vamos a hacer el diagrama de actividad más representativo de la aplicación, se corresponde con grabar una actividad.

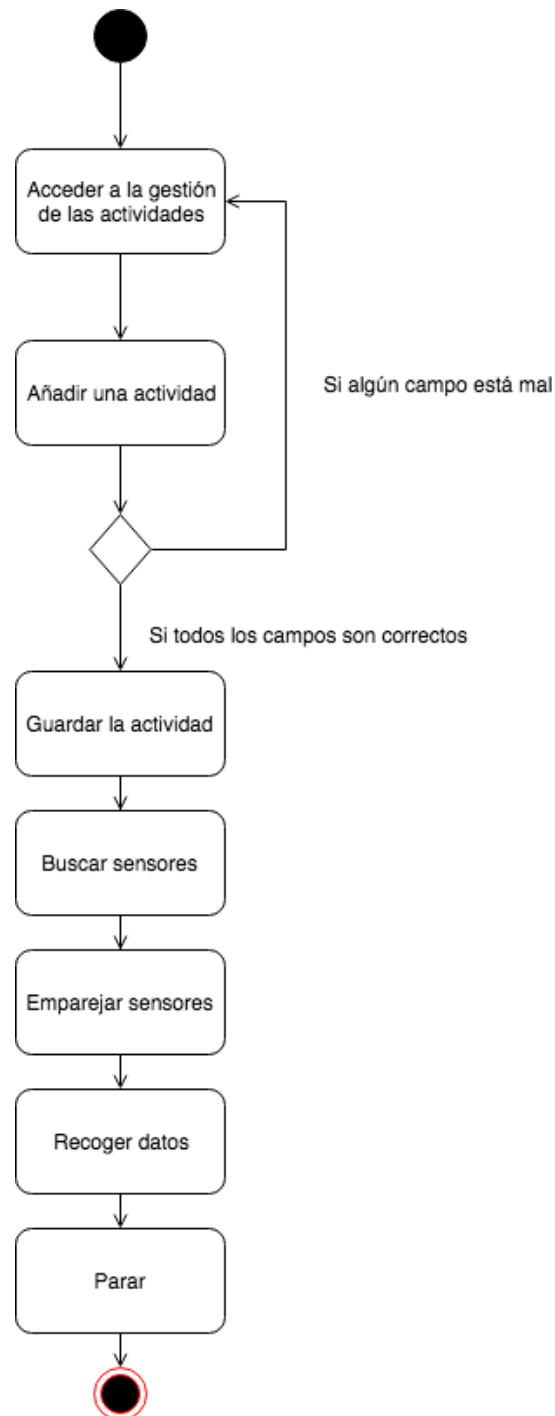


Figura 22: Diagrama de actividad grabar actividad.

Nota: el emparejamiento de sensores se realiza por parte del usuario. Primero se buscan los dispositivos disponibles y a continuación, se elige el sensor y se pulsa en emparejar.

7.4. Aspectos importantes de la implementación

La aplicación no sigue ninguna arquitectura concreta. Se diferencian la capa del modelo y la capa de la vista. La vista está formada por todos los layouts de Android y el modelo por las actividades de Android.

Diagrama

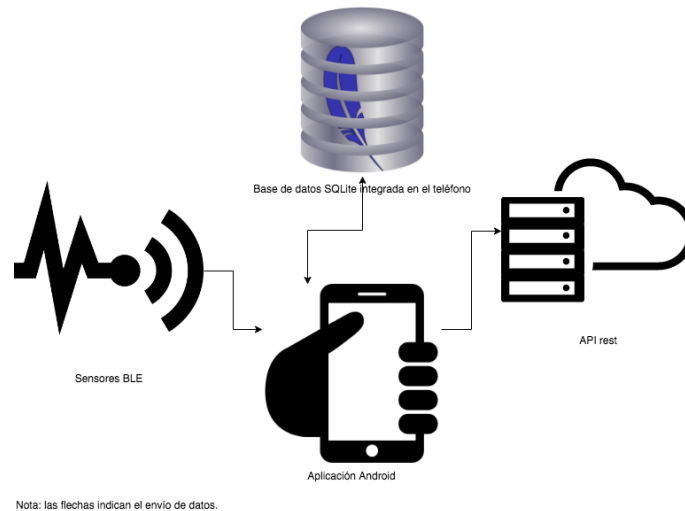


Figura 23: Arquitectura aplicación Android.

Estructura de clases

El código de la aplicación está organizado en tres grandes grupos:

- **Actividades:** en este paquete se encuentran todas las actividades que forman la aplicación de Android. Esto se corresponde con cada una de las funcionalidades de la aplicación, es decir, actividades, medicamentos y recogida de los datos emitidos por los sensores.
- **Adaptadores:** son principalmente dos, uno para las actividades y otro para los medicamentos. Permiten la creación de listas personalizadas en Android. De esta forma podemos mostrar de una forma sencilla e intuitiva toda la información relacionada con cada elemento, así como editar y borrar cada uno de ellos.
- **Clases:** aquí podemos encontrar gran variedad. Por un lado están las clases que representan cada objeto dentro de la aplicación, como son el objeto que representa a un medicamento y el que representa a una actividad. Por otro lado están los servicios que gestionan la conexión bluetooth con los sensores. Por último, se encuentra la clase que conecta la aplicación con la API rest.

Entrando un poco más en detalle, vamos a explicar cómo funcionan los adaptadores, ambos funcionan igual, ya que controlan una funcionalidad parecida. Los adaptadores, son una forma que ofrece Android de personalizar las listas. En nuestro caso tenemos dos listas principales, la de medicamentos y la de actividades, el adaptador se encarga de mostrar toda la información de cada actividad y medicamento en concreto, de una forma sencilla para su lectura. A su vez, el adaptador permite también

llevar a cabo las operaciones de editar y borrar cada elemento de la lista.

Por otro lado, todas las actividades funcionan de una forma similar. Las actividades es una funcionalidad que ofrece Android. Cada una de ellas se corresponden con cada una de las pantallas que se ven a lo largo del uso de la aplicación. Cada una cuenta con un layout, que define la forma en la que se va a ver todo en esa pantalla. A su vez, mediante mecanismos de Android, se pueden coger datos del layout, esto es, detectar pulsaciones en botones y recoger los datos de los diferentes formularios.

El último elemento a destacar son las llamadas a la API Rest. Se hacen mediante una librería de Android (Volley) Se utilizan mensajes JSON para transmitir toda la información, tanto de la aplicación a la API como viceversa.

Base de datos

Entraremos en más en detalle en los elementos que forman la aplicación. El almacenamiento de datos dentro del teléfono se realiza gracias a SQLite ("SQLite es una librería escrita en lenguaje C que implementa un manejador de base de datos SQL embebido.") En ella se guardan todos los datos del sensor que aún no han sido enviados al servidor, una vez se envían se borran de la base de datos local. Los medicamentos y actividades, disponen de un campo que indica si se han enviado al servidor o no, porque a diferencia de los datos de los sensores, no se borran nunca del teléfono. El diagrama entidad relación es el siguiente:

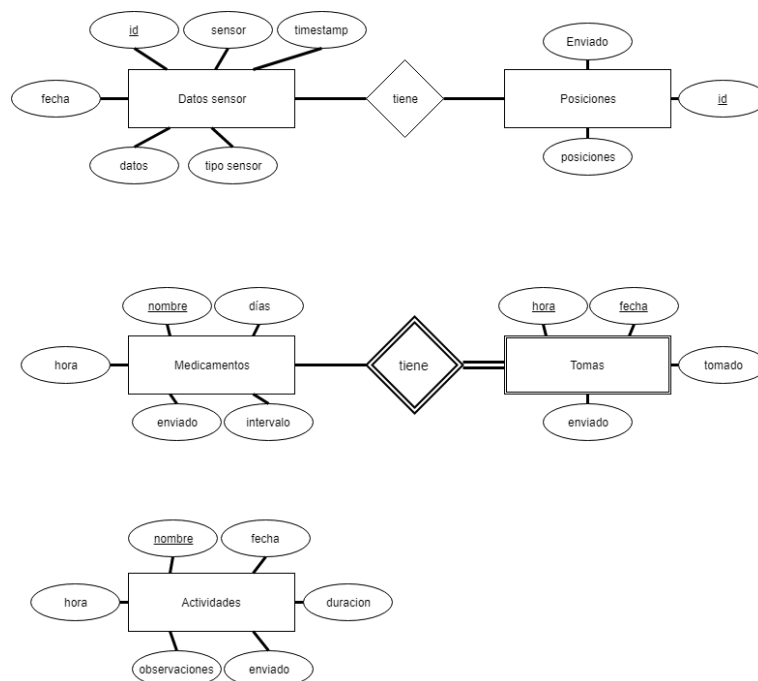


Figura 24: Diagrama entidad-relación de la base de datos local.

Cada entrada de datos sensor tendrá un campo posiciones que sera el id correspondiente de la tabla posiciones que representa las posiciones en las que están situados los sensores. Para cruzar los datos de las actividades y los datos del sensor se usan la fecha y la hora.

Conexión con los sensores

La conexión con cada tipo de sensor se hace de forma diferente. Ambas se realizan con librerías que hemos encontrado en GitHub [17] y hemos adaptado a nuestras necesidades. La aplicación permite conectar varios sensores de Texas Instruments a la vez, mientras que sólo permite conectar un sensor del tipo Hexiwear. Para mantener la conexión activa con los sensores, aunque la aplicación no esté en primer plano, hemos utilizado los servicios de Android. De tal forma que aunque la aplicación esté en multitarea, o el móvil se encuentre bloqueado, los sensores siguen conectados y enviando datos al teléfono. Con esto, conseguimos que el paciente no tenga que tener el móvil siempre a mano ni estar pendiente de él mientras realiza cada actividad. Debido a que usamos BLE, que es bluetooth de bajo consumo, la conexión con los sensores se dificulta, ya que se pierde con mayor facilidad, por ello los servicios ayudan a mantener en todo momento la conexión activa.

8. Aplicación web

8.1. Descripción de la funcionalidad

La aplicación web permite al médico encargado de supervisar a un paciente de Parkinson seguir la evolución de éste. La aplicación web muestra tanto la información recogida por las pulseras con sensores BLE correspondiente a la ejecución de las distintas actividades por parte del paciente, como la información recogida en la bitácora de medicamentos. De esta forma el médico puede conocer cuando el paciente ha dejado de tomarse la medicación y tener esto en cuenta a la hora de visualizar los datos.

La aplicación web ha sido diseñada para ser lo más simple e intuitiva posible y mostrar sólo la información necesaria para que el médico pueda interpretar los datos recogidos por los sensores.

A continuación se muestran las tres vistas con las que cuenta la aplicación web, a excepción de la pantalla de inicio de sesión que se ha excluido de esta sección por su simpleza:

Pantalla principal

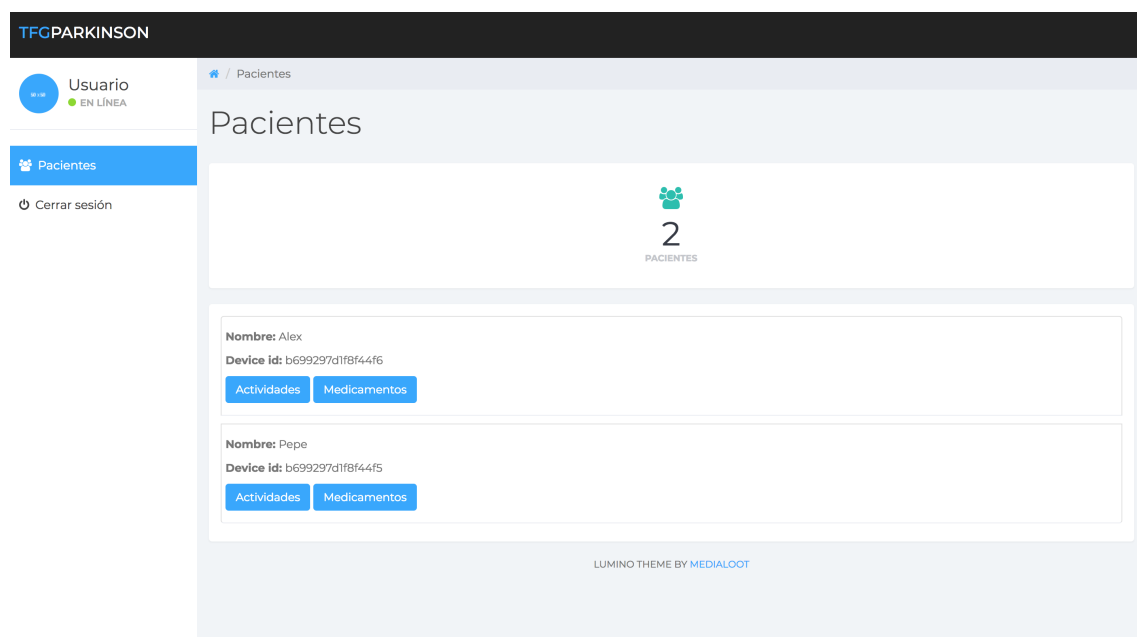


Figura 25: Pantalla principal de la aplicación web.

En la pantalla principal se muestra el número total de pacientes que están registrados en el sistema y un listado de estos con su nombre e identificador además de dos botones que nos permiten acceder a la información de las actividades y la medicación de cada paciente.

Pantalla de actividades

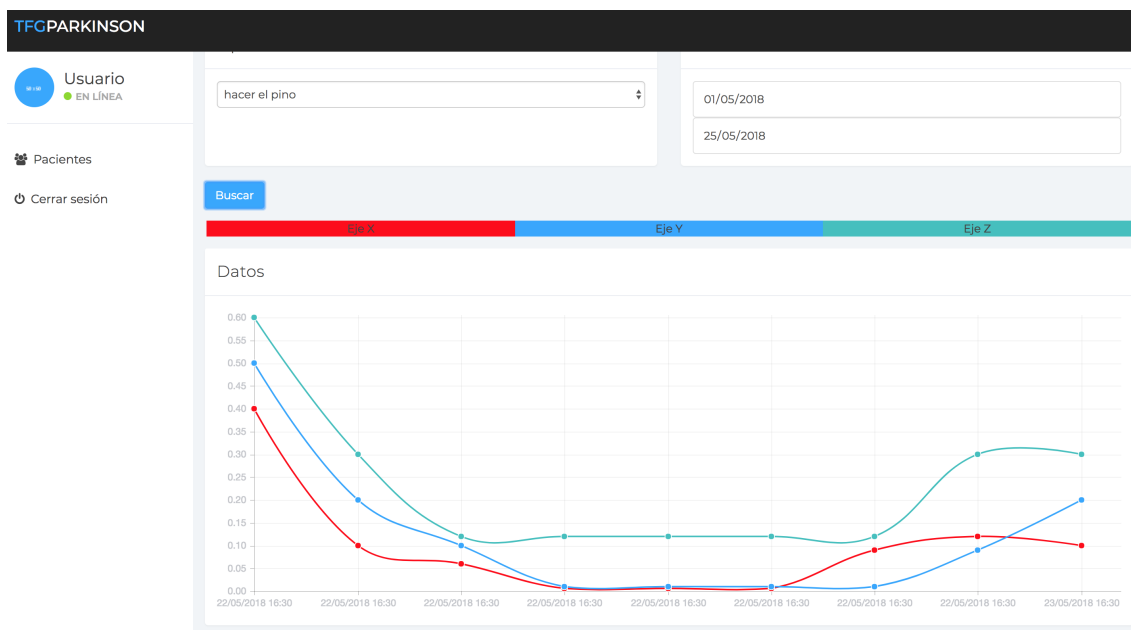


Figura 26: Pantalla de actividades de un paciente de la aplicación web.

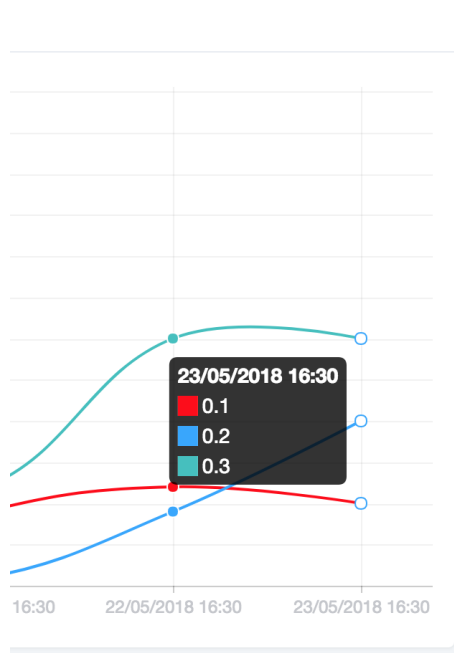


Figura 27: Pantalla de detalle dato sensor de una actividad.

En la pantalla de actividades se muestra la información recopilada por las pulseras con sensores BLE mientras que el paciente ejecutaba las actividades. Para visualizar los datos se debe seleccionar, mediante un desplegable, la actividad que se quiere visualizar y las fechas entre las que se quiere ver esa actividad.

Una vez seleccionados los desplegables se mostrará una gráfica por eje de cada sensor disponible en la que se compararán los datos disponibles de actividades de ese tipo realizadas entre las fechas seleccionadas. Esto permitirá al médico ver de una forma clara cómo ha evolucionado la forma en la que el paciente realiza las actividades.

Ésta es una de las pantallas que más información aporta al médico, ya que es aquí donde puede ver de una forma gráfica y sencilla la evolución del paciente.

Pantalla de medicación

Figura 28: Pantalla de medicación de un paciente de la Aplicación web.

La pantalla de medicación hace uso de la bitácora de medicamentos para proporcionar al médico información sobre las tomas de la medicación del paciente. El médico tan solo tiene que seleccionar el nombre del fármaco que quiere consultar y el rango de fechas y se le mostrarán las fechas en las que el paciente debía haber tomado esa medicina y no la tomó. Esta información sirve tanto para aportar un contexto a los datos mostrados en la ventana de actividades, como para detectar si hay algún problema con el paciente o con la medicación que tiene asignada.

8.2. Diagramas de casos de uso

A continuación se muestran los diagramas de casos de uso correspondientes a la aplicación web:

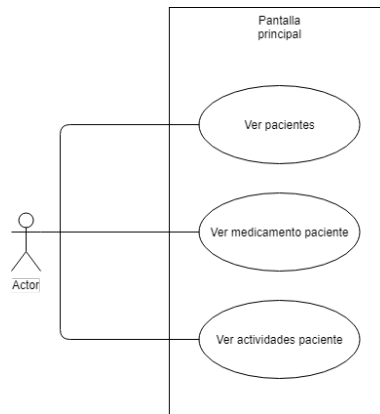


Figura 29: Caso de uso - Pantalla principal de la Aplicación web..

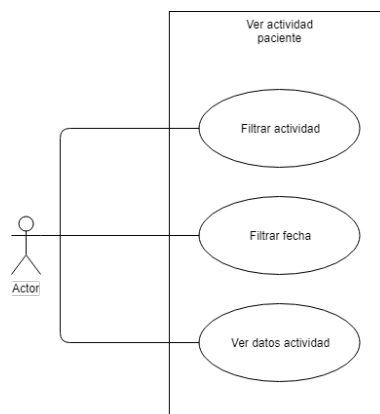


Figura 30: Caso de uso - Pantalla de actividades de un paciente de la Aplicación web.

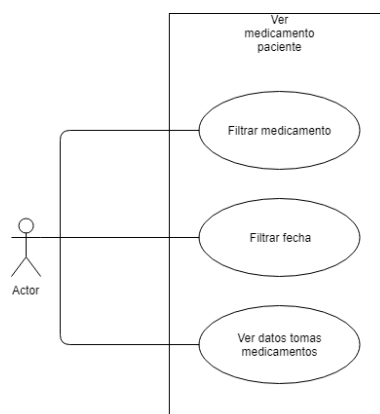


Figura 31: Caso de uso - Pantalla de medicación de un paciente de la Aplicación web.

8.3. Diagramas de actividad

A continuación se muestra el diagrama de actividad de la principal actividad de la aplicación web, ver los datos de las actividades de un paciente:

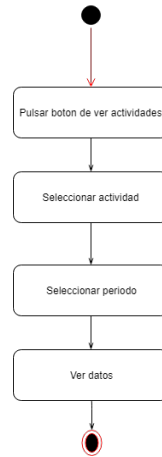


Figura 32: Diagrama actividad - Ver datos de la actividad de un paciente.

8.4. Aspectos importantes de la implementación

La aplicación web está desarrollada usando principalmente HTML, Bootstrap y JavaScript, aunque hacemos uso de PHP para gestionar las sesiones y hacer las llamadas al API REST.

Todos los datos mostrados en la aplicación se recogen a través de JavaScript haciendo uso del API REST. Para evitar problemas de CORS (Intercambio de Recursos de Origen Cruzado), que surgen al realizar peticiones HTTP a un dominio diferente del de la aplicación web, hemos implementado una pagina PHP que recibe mediante JavaScript la consulta que se quiere realizar al API REST y la realiza desde el backend devolviendo los datos tal cual para su posterior procesado mediante el propio código JavaScript.

Hemos seleccionado PHP para implementar la parte de backend ya que el uso que le damos es mínimo y no creemos que sea necesario el uso de otras tecnologías usadas en proyectos de mayor tamaño.

El hecho de usar el API REST para recuperar los datos y el uso tan reducido que hacemos del backend nos permite poder migrar la aplicación web de una forma muy sencilla. En caso de que alguien quisiese usar nuestra aplicación con una tecnología distinta a PHP tan solo tendría que crear una pagina que realizase llamadas HTTP al API REST y modificar en JavaScript la ruta de las llamadas.

9. Servicios del servidor

En esta sección se describen los diferentes servicios ejecutados en la parte del servidor que sirven principalmente para el envío bidireccional, almacenamiento y procesamiento de los datos obtenidos de la grabación de las actividades realizadas por los pacientes a través de las pulseras con sensores.

9.1. API REST

Para interactuar con la base de datos del servidor decidimos usar un API REST. Esto nos aporta una capa de separación entre el cliente y el servidor lo que aumenta la escalabilidad del proyecto y nos permite usar el mismo interfaz tanto para la aplicación Android como para la aplicación web y al usar el protocolo HTTP nos permitiría, en un futuro, si fuese necesario, añadir mas componentes de una forma sencilla.

9.1.1. Detalles técnicos

El API REST está implementada usando Eve, un framework opensource basado en Python. Eve nos permite implementar y desplegar el API de una forma rápida y sencilla usando apenas 3 lineas de código. Tan solo es necesario hacer un programa en Python que se encarga de levantar el API REST pasándole como parámetro todas las variables necesarias para establecer las peculiaridades del API REST, en nuestro caso dirección IP y puerto en el que queremos que escuche, y tener definido un archivo de configuración con el esquema de datos (apéndice 14.1) y métodos permitidos, en nuestro caso POST y GET.

9.1.2. Esquema de datos

El esquema JSON (apéndice (14.1) definido para establecer la estructura de las llamadas al API, se corresponde principalmente con las tablas de la base de datos local de la aplicación Android. Se añade a cada tabla un campo identificador para cada paciente, este campo se coge automáticamente del dispositivo Android en el que se esté usando la aplicación.

Además, el esquema cuenta con una tabla para usuarios de la aplicación web que correspondería a los médicos que tengan acceso a los datos de los pacientes supervisados y otra a información del paciente asociada a su identificador.

9.2. Almacenamiento de datos

Para llevar a cabo la tarea del almacenamiento de datos hemos usado MongoDB ya que es una de las bases de datos noSQL más usada, con una gran comunidad por detrás, que además ofrece toda la flexibilidad que caracteriza las bases de datos noSQL y que por ello nos es de utilidad para este proyecto. Además, MongoDB está totalmente integrada en el framework que después usamos para implementar el API REST.

El esquema de datos usado en la base de datos del servidor se corresponde totalmente con el del API REST ya que es a través de ésta como se realizan las inserciones y consultas de datos.

9.3. Servidor HTTP

Hemos desplegado nuestro propio servidor web para servir la aplicación web de visualización de datos. El papel principal del servidor web es gestionar las diferentes sesiones ya que todos los datos que se muestran en la web se recogen mediante JavaScript haciendo uso del API REST.

Además, el uso del servidor web nos permite evitar los problemas relacionados con CORS (Intercambio de Recursos de Origen Cruzado) que surgen al estar la aplicación web y el API REST en dominios distintos, ya que realizamos las consultas HTTP usando PHP desde el lado del servidor de la aplicación.

9.3.1. Detalles técnicos

Como servidor HTTP para servir la aplicación web hemos usado Apache Server sobre Ubuntu16.04 junto con PHP 7.0.

9.4. Despliegue de la infraestructura

Para facilitar el despliegue de la infraestructura se ha automatizado el proceso de despliegue usando Docker y un script en shell. Esto nos permite desplegar la infraestructura en cualquier tipo de sistema, independientemente del SO y de una forma rápida y sencilla.

9.4.1. Infraestructura sobre Docker

Para desplegar la infraestructura sobre Docker hemos creado dos imágenes propias, partiendo de la imagen oficial de Ubuntu 16.04. En una de ellas se instala y despliega todo lo necesario para el API REST y en la otra se instala y se despliega todo lo necesario para el servidor web. Además, hacemos uso de la imagen oficial de MongoDB para levantar la base de datos, lo que da lugar a tres contenedores de Docker enlazados entre sí como muestra el siguiente diagrama:

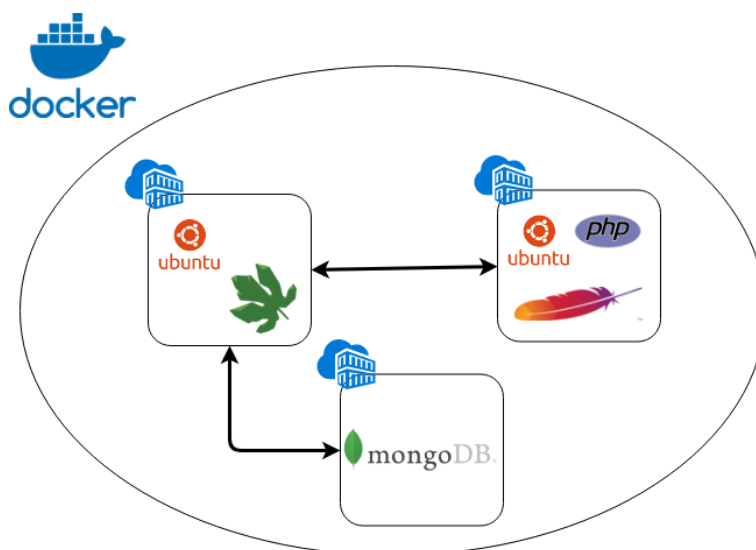


Figura 33: Diagrama de interacción de los contenedores de Docker.

9.5. Procesamiento de datos en el servidor

Pese a que por dificultades ajenas al proyecto no hemos podido realizar una fase de interacción con usuarios reales para recopilar datos de enfermos de Parkinson con los que poder trabajar y aplicar las técnicas de datos necesarias para poder extraer información de utilidad, hemos implementado un programa en Python que se encarga de procesar los datos almacenados, por el API REST, en la base de datos remota para producir datos que pudiesen ser usados para entrenar algoritmos de machine learning de una forma sencilla.

10. Conclusiones

El trabajo lo comenzamos en julio del año 2017. El objetivo inicial asociado al Parkinson era desarrollar una aplicación móvil que permitiera la monitorización de algún fenómeno con cierta relevancia médica. Al no conseguir el apoyo médico, optamos por seguir la línea del desarrollo de un sistema más genérico que permite la supervisión remota del paciente. Nuestro sistema está inspirado en el que se puede ver en el artículo [1], aunque tiene ciertas diferencias. La supervisión del paciente se facilita mucho debido a que se puede hacer de forma remota, en todo momento el médico y cuidador pueden disponer de los datos del paciente a través de una página web.

El sistema puede tener varios usos, en los que hemos pensado principalmente son los siguientes:

- **Monitorización de tablas de ejercicios:** el paciente puede hacer diariamente los ejercicios que le mande su médico y se almacenan todos los datos recogidos por los sensores, así como las posibles observaciones que pueda hacer el paciente después de realizar la actividad.
- **Monitorización del efecto de la medicación:** los sensores recogen datos tanto antes como después de la toma de la medicación. Esto permite ver el efecto que está teniendo la medicación y podría permitir la médico ver si la medicación es la correcta.
- **Bitácora de la medicación:** sirve para llevar un control de las pastillas que se toma el paciente, así como de si se las ha tomado o no. A su vez también se lleva un control del seguimiento de la tabla de ejercicios.

Nuestro trabajo sigue la línea de los que vimos en los artículos del estado del arte. Aunque, en nuestro caso, hemos querido centrarnos más en otros aspectos y no tanto en el análisis de datos posterior. El sistema ayuda al paciente a recordar la toma de sus pastillas y le permite una vez tomada la pastilla grabar la actividad con los sensores, de forma que posteriormente el médico puede tener una visión completa del efecto que tiene la medicación. Mientras que en la mayoría de artículos se centraban en el ejercicio supervisado, nuestro sistema deja ese apartado libre, pudiendo el paciente realizar su tabla de ejercicios o ejercicio libre. Todos los datos recogidos, pueden ser visualizados por el médico de forma sencilla y filtrando la información dependiendo del tipo de datos que quiera ver. De esta forma, de manera remota, puede tener el control de todos sus pacientes y puede tomar decisiones en cualquier momento. De esta forma se quiere seguir la línea de objetividad y continuidad que planteaban muchos artículos, cómo por ejemplo el [2]

Todos los datos que se recogen, son almacenados en el sistema, primero en el móvil y luego en el servidor, donde son utilizados para ser visualizados por parte del médico a través de la página web. A su vez, se guardan de tal forma que en el futuro podrían ser analizados en mayor profundidad e incluso se podría realizar machine learning con ellos.

Como conclusión final, a pesar de la falta de personas que prueben nuestra aplicación, se ha desarrollado un trabajo muy completo y cerrado. Es completamente

funcional y se podría poner en funcionamiento en cualquier momento y a partir de esto ampliar y seguir con la idea inicial del proyecto.

11. Conclusions

The work was started in the last July. The initial objective was to develop a smartphone application for monitoring some phenomenon with medical relevance. We didn't get medical support, so we followed the line of developing a more generic system which allows patient's remote supervision. Our system is inspired in the one which can be seen in the article [1], although it has some differences. The patient supervision is facilitated because it can be done remotely. Through a web page the doctor and the carer can access the patient's data.

The system can be used in different ways, these are the main ones:

- **Monitoring of activities tables:** the patient can perform the exercises that his doctor recommends. The data, received from the sensors, are stored. Any observation from the patient about the activity can be stored too.
- **Monitoring of the medicine effect:** the application stores data before and after taking the medicine. This can be used by the doctor to check if the medicine is working well.
- **Medication log:** it can be used to see if the patient is taking all the medicines and also it allows to follow all the activities.

Our work follows the line of some of the articles we read. Although, we focused more in other aspects and not as much in the data analysis. The system helps the patient to remind the medicine taking and it allows to record the activity after that. In this way the doctor can have a complete vision of the medicine effect. While in the majority of the articles, the systems proposed required supervision by a doctor or a carer, our system is designed to be used without it. All the data can be seen and filtered by the doctor. In this way, the doctor can remotely supervise the patient. So we can follow the objective and continuation line that some articles follow, for example [2].

Data from the sensors is stored first in the smartphone, then in the server, where it can be displayed through a web application.

Finally, although our system has not been tested by patients, we have developed a complete work. It is functional and can be used in any moment.

12. Trabajo futuro

El trabajo futuro se centraría fundamentalmente en tres de los elementos fundamentales del proyecto. Mientras que aspectos como la API o el despliegue de la arquitectura han quedado cerrados en esta iteración.

12.1. Aplicación Android

La aplicación podría ser completada añadiendo los siguientes elementos:

- Gráficas en tiempo real con los datos que son recogidos por los sensores.
- Gestión de los temblores, que en un primer momento se planteó, pero que con el cambio del rumbo del TFG se descartó. La gestión de temblores consiste en permitir al paciente indicar cuándo está teniendo un temblor o cuándo lo tuvo y de esta forma poder contrastar esos datos con datos de otras actividad. Consiguiendo así diferenciar cuándo el paciente está teniendo un temblor de cuando está realizando otra actividad.
- Recordatorios personalizables para las pastillas. Ahora mismo la aplicación sólo manda una notificación a la hora en el día que se tiene que tomar la pastilla. La idea es que el paciente pudiera indicar si quiere que el terminal vibre o suene cuando salte la notificación, ya que ahora mismo sólo se muestra.
- Varias vistas de tal forma que el paciente vea una parte de la aplicación y su cuidador pueda acceder a otros datos.
- Que la comunicación con la pulsera sea bidireccional, de forma que se puedan enviar notificaciones a la pulsera. Esto sería útil para que el recordatorio de toma de la pastilla pudiera ser visto en la pantalla de la pulsera, de esta forma el paciente no necesitaría tener el móvil siempre cerca. Además, se podría ver la posibilidad de que la pulsera vibrara con cada recordatorio, de esta forma el paciente sabría en todo momento qué pastilla se tiene que tomar.
- Utilizar vibración o sonidos dentro de las notificaciones o a la hora de alcanzar valores extraños por parte de los sensores, como por ejemplo que el sensor se ha desconectado o que el sensor está devolviendo valores erróneos.
- Por último, el diseño de la aplicación es bastante básico. Pero se podría personalizar para adaptarlo a personas mayores o con problemas para el manejo de pantallas táctiles de pequeñas dimensiones.

12.2. Aplicación Web

La web podría ser completada añadiendo los siguientes elementos:

- Permitir el registro de usuarios, tanto de médicos como de pacientes.
- Ahora mismo la web está orientada a la visualización de datos por parte del médico, pero quizás en un futuro sería interesante que el paciente la pudiera usar para ver sus datos, así como recomendaciones dadas por su médico.

- Actualmente sólo se muestran datos de actividades y medicamentos por paciente, en el futuro podrían mostrarse datos cruzados de pacientes e intentar así mejorar los diagnósticos.
- Permitir al usuario elegir el formato de cada uno de los gráficos que se muestran, así como seleccionar los datos que aparecen en ellos.

12.3. Estudio de los datos

Todos los datos recogidos en un primer momento iban a ser utilizados para aplicarles machine learning y así poder obtener predicciones acerca de la evolución del paciente. Todo el sistema está orientado para obtener los datos de forma sencilla y poder analizarlos en cualquier momento.

El análisis podría ir encaminado a supervisar la realización de los ejercicios, detectando anomalías y temblores. O, con los sensores en los tobillos, podría utilizarse para monitorizar la congelación del paso.

13. Aportaciones

13.1. Christian González

Este proyecto está caracterizado porque tanto mi compañero como yo hemos intentado trabajar en grupo e intentar que ambos aportásemos nuestro granito de arena a cada una de las partes de este proyecto. Bien es cierto, que para agilizar el desarrollo y ser capaces de cumplir con los plazos marcados por nosotros mismos y nuestros directores de proyecto, hemos tenido que paralelizar tareas repartiendo el trabajo entre ambos.

Teniendo en cuenta lo anterior creo que mis aportaciones a las diferentes partes del proyecto se podrían enumerar de la siguiente forma:

13.1.1. Estado del arte

Antes de comenzar a realizar este proyecto los directores del proyecto nos recomendaron la lectura de varios artículos científicos que hablaban sobre experimentos similares al nuestro. En los que de una forma u otra se monitorizaba a pacientes con distintos síntomas de la enfermedad del Parkinson. Ya fuese para intentar identificar, usando técnicas de análisis de datos, las actividades que se estaban realizando entre un conjunto de actividades dadas o tan solo intentar identificar si un paciente sufría síntomas de la enfermedad o el grado de estos síntomas. La lectura en profundidad de estos artículos nos permitió tanto ser conscientes del contexto en el que se encuadraba nuestro proyecto como realizar el apartado de Estado del Arte de esta memoria.

13.1.2. Aplicación Android

Dentro de la aplicación Android creo que mis aportaciones individuales destacan más en el diseño e implementación de la base de datos local, así como en la integración con los sensores BLE mientras que en la parte de diseño e implementación de la interfaz de usuario mis aportaciones han sido menores.

13.1.3. Aplicación web

La aplicación web es una de las partes que está realizada casi en su totalidad trabajando en grupo por lo que creo que tanto mi compañero como yo hemos aportado en prácticamente la totalidad del trabajo implementando las partes del backend necesarias en PHP y todas las llamadas al API REST para recuperar y mostrar los datos en el cliente, dado que la parte de diseño esta inspirada en plantillas online.

13.1.4. API REST

Como el apartado anterior el API REST ha sido diseñado he implementado trabajando en grupo por lo que mis aportaciones cubren prácticamente el total del trabajo en este apartado, desde el diseño del esquema de datos a la implementación de esta usando el framework Eve.

13.1.5. Despliegue de la infraestructura

Esta sección es probablemente la sección en la que mis aportes personales destacan más. Entre estas aportaciones se encuentran la de la creación de las imágenes de Docker usadas para desplegar los contenedores con el servidor web y el API REST y la implementación del script de despliegue que permite la automatización del despliegue de la infraestructura.

13.1.6. Procesamiento de datos

El programa que se encarga de procesar los datos almacenados, por el API REST, en la base de datos remota es otra aportación grupal cuya lógica desarrollamos y pensamos mi compañero y yo trabajando en grupo.

13.1.7. Otras aportaciones

Aparte de las contribuciones a los apartados principales del proyecto, creo que es necesario mencionar el trabajo de pruebas realizadas para asegurar el correcto funcionamiento de cada parte del proyecto por separado y de la integración de todas ellas, así como las aportaciones a la realización de la propia memoria del proyecto

13.2. Alejandro Huertas Herrero

Mi aportación al proyecto ha sido de un 50 % aproximadamente. Mi compañero y yo nos hemos repartido las tareas más o menos por igual. A continuación, voy a enumerar las partes en las que he trabajado, obviamente mi compañero también ha trabajado, pero las voy a describir desde mi punto de vista y trabajo realizado por mí.

13.2.1. Lectura de artículos

Al comienzo del TFG, por el mes de julio de 2017, empezamos a leernos artículos científicos, relacionados con el tema del trabajo. En mi caso, me leí un total de siete artículos. De cada uno de ellos, hice un resumen extrayendo lo más útil para nuestro trabajo. Todos los artículos que analicé están mencionados en el apartado de estado del arte.

13.2.2. Aplicación Android

Me he ocupado de todo lo relativo al diseño de la aplicación, así como de la conexión de la aplicación con la API Rest. Decidí hacer esto yo, porque durante mi primer trabajo en prácticas estuve trabajando con Android y sabía muy bien como manejarlo. El diseño de la aplicación lo he ido haciendo yo de acuerdo tanto con mi compañero como con nuestros directores. Aunque en menor medida, también he participado en la conexión de la aplicación con los sensores, sobre todo, en la conexión de los sensores Hexiwear. Otra tarea muy importante ha sido la búsqueda de toda la información necesaria para conectar los sensores a la aplicación. Nos pasamos muchísimo tiempo buscando por Internet cómo se conectaban, hasta que mi compañero dio con una solución viable.

13.2.3. Aplicación Web

Cuando se nos propuso hacer la aplicación web para cerrar el TFG, me puse a buscar plantillas que fueran sencillas y tuvieran todo lo necesario. Acabé encontrando una plantilla que utiliza Bootstrap. Modifiqué la plantilla para dejar el diseño como lo necesitábamos nosotros. También me he encargado junto a mi compañero de realizar todo el Javascript necesario, así como el PHP que nos permite comunicarnos con la API Rest para obtener la información necesaria. La web la estamos haciendo los dos juntos, ninguno está aportando más que el otro, la hacemos a través de Skype y cuando podemos quedamos en persona.

13.2.4. API Rest

Cuando pensamos en la idea de guardar los datos de la aplicación en un servidor, a los dos se nos vino a la cabeza la idea de utilizar MongoDB como sistema de almacenamiento. En ese momento buscamos si había algún framework disponible para Python que usara MongoDB, dimos con EVE. La verdad que es muy sencillo de utilizar, entre los dos elaboramos el fichero de settings que necesita para poder funcionar. También entre los dos, hicimos un script en Python que permite juntar los datos recogidos por cada sensor, de forma que se puedan procesar todos los datos de una persona a la vez.

13.2.5. Memoria

La memoria la hemos dividido a partes iguales, cada uno ha hecho la parte correspondiente a cada parte en la que más esfuerzo ha puesto. En mi caso, he hecho las partes relacionadas con Android, así como la descripción de los sensores de Texas y he participado en el resto de partes aportando o corrigiendo ciertas cosas.

13.2.6. Organización

Aunque nos hemos organizado juntos, yo he puesto más hincapié en utilizar aplicaciones como Trello, para poder llevar un control de todo lo que había que hacer y sobre todo de lo que tenía que hacer cada uno de nosotros.

14. Apéndices

14.1. Esquema de datos API REST

```
1 {'temblores': {
2     'additional_lookup': {
3         'url': 'regex("[\w]+")',
4         'field': 'id'
5     },
6     'schema': {
7         'id': {
8             'type': 'integer'
9         },
10        'duracion': {
11            'type': 'integer'
12        },
13        'observaciones': {
14            'type': 'string'
15        },
16        'timestamp_inicio': {
17            'type': 'integer'
18        },
19        'device_id': {
20            'type': 'string'
21        }
22    }
23 },
24 'datos_sensor': {
25     'additional_lookup': {
26         'url': 'regex("[\w]+")',
27         'field': 'posiciones'
28     },
29     'schema': {
30         'id': {
31             'type': 'integer'
32         },
33         'sensor': {
34             'type': 'string'
35         },
36         'posiciones': {
37             'type': 'integer'
38         },
39         'db_timestamp': {
40             'type': 'string'
41         },
42         'app_timestamp': {
43             'type': 'string'
44         },
45         'datos': {
```

```

46         'type': 'string'
47     },
48     'device_id': {
49         'type': 'string'
50     },
51     'tipo_sensor': {
52         'type': 'string'
53     }
54 }
55 },
56 'posiciones': {
57     'additional_lookup': {
58         'url': 'regex("[\w]+")',
59         'field': 'id'
60     },
61     'schema': {
62         'id': {
63             'type': 'integer'
64         },
65         'posiciones': {
66             'type': 'string'
67         },
68         'device_id': {
69             'type': 'string'
70         }
71     }
72 },
73 'medicamentos': {
74     'additional_lookup': {
75         'url': 'regex("[\w]+")',
76         'field': 'nombre'
77     },
78     'schema': {
79         'nombre': {
80             'type': 'string'
81         },
82         'dias': {
83             'type': 'string'
84         },
85         'hora': {
86             'type': 'string'
87         },
88         'minutos_descartar': {
89             'type': 'integer'
90         },
91         'device_id': {
92             'type': 'string'
93         }
94     }

```

```

95     },
96     'actividades': {
97         'additional_lookup': {
98             'url': 'regex("[\w]+")',
99             'field': 'nombre'
100     },
101     'schema': {
102         'nombre': {
103             'type': 'string'
104         },
105         'intervalo': {
106             'type': 'integer'
107         },
108         'hora': {
109             'type': 'string'
110         },
111         'device_id': {
112             'type': 'string'
113         },
114         'observaciones': {
115             'type': 'string'
116         },
117         'fecha': {
118             'type': 'string'
119         }
120     }
121 },
122 'usuario': {
123     'additional_lookup': {
124         'url': 'regex("[\w]+")',
125         'field': 'nombre'
126     },
127     'schema': {
128         'nombre': {
129             'type': 'string',
130             'unique': True
131         },
132         'password': {
133             'type': 'string'
134         }
135     }
136 },
137 'paciente': {
138     'additional_lookup': {
139         'url': 'regex("[\w]+")',
140         'field': 'nombre'
141     },
142     'schema': {
143         'nombre': {

```

```

144         'type': 'string',
145         'unique': True
146     },
147     'device_id': {
148         'type': 'string',
149         'unique': True
150     }
151 }
152 },
153 'tomas': {
154     'additional_lookup': {
155         'url': 'regex("[\w]+")',
156         'field': 'medicamento'
157     },
158     'schema': {
159         'medicamento': {
160             'type': 'string'
161         },
162         'hora': {
163             'type': 'string'
164         },
165         'fecha': {
166             'type': 'string'
167         },
168         'tomado': {
169             'type': 'string'
170         },
171         'device_id': {
172             'type': 'string',
173             'unique': True
174         }
175     }
176 }
177 }

```

14.2. Repositorios GitHub

Todo el código del sistema se encuentra en tres repositorios de GitHub, son los siguientes:

- Aplicación Android: <https://github.com/al3xhh/TFGParkinsonAplicacion>
- Aplicación Web: <https://github.com/al3xhh/TFGParkinsonAplicacion>
- API rest y docker: <https://github.com/christian7007/TFGParkinsonAPI>

Referencias

- [1] shyamal patel, konrad lorincz, richard hughes, nancy huggins, john h. growdon, matt welsh, paolo bonato, senior member, ieee. "analysis of feature space for monitoring persons with parkinson's disease with application to a wireless wearable sensor system"
- [2] walter maetzler, josefa domingos, karin srulijes, joaquim j. ferreira, bastiaan r. bloem. "quantitative wearable sensors for objective assessment of parkinson's disease"
- [3] baukje dijkstra, wiebren zijlstra, erik scherder, yvo kamsma. "detection of walking periods and number of steps in older adults and patients with parkinson's disease: accuracy of a pedometer and an accelerometry-based method"
- [4] shyamal patel, konrad lorincz, richard hughes, nancy huggins, john growdon, david standaert, metin akay, fellow, ieee, jennifer dy, matt welsh, member, ieee, and paolo bonato, senior member, ieee. "monitoring motor fluctuations in patients with parkinson's disease using wearable sensors"
- [5] C.L. Pulliam, S.R. Eichenseer, C.G. Goetz, O. Waln, C.B. Hunter, J. Jankovic, D.E. Vaillancourt, J.P. Giuffrida, D.A. Heldman Continuous in-home monitoring of essential tremor
- [6] wikipedia: https://es.wikipedia.org/wiki/enfermedad_de_parkinson
- [7] wikipedia: <https://es.wikipedia.org/wiki/hipocinesia>
- [8] wikipedia: <https://en.wikipedia.org/wiki/dyskinesia>
- [9] wikipedia: https://en.wikipedia.org/wiki/parkinsonian_gait
- [10] wikipedia: https://en.wikipedia.org/wiki/support_vector_machine
- [11] texas instruments: <http://www.ti.com/product/cc2650/description>
- [12] wikipedia texas instruments: http://processors.wiki.ti.com/index.php/cc2650_sensortag_user's_guide
- [13] aplicación texas instruments: <https://play.google.com/store/apps/details?id=com.ti.ble.sensortag>
- [14] national institute of neurological disorders and stroke: https://espanol.ninds.nih.gov/trastornos/parkinson_disease_spanish.htm
- [15] asociación parkinson madrid: <https://www.parkinsonmadrid.org/el-parkinson/el-parkinson-definicion/>
- [16] tfg anterior: reconocimiento de actividad mediante pulsera con sensores"
- [17] github con ejemplo conexión: <https://github.com/silvia-valdez/AndroidProjects-Java/tree/master/Examples/MultiBLEConnectionApp>